

Chapter 9

Layer 3 VPN Configuration Examples

This chapter provides the following examples of Layer 3 virtual private network (VPN) configurations:

Configure a Simple Full-Mesh VPN Topology on page 128

Configure a Full-Mesh VPN Topology with Route Reflectors on page 142

Configure a Hub-and-Spoke VPN Topology on page 142

Configure an LDP-over-RSVP VPN Topology on page 158

Configure an Application-Based Layer 3 VPN Topology on page 172

Configure an OSPF Domain ID for a Layer 3 VPN on page 177

Configure Overlapping VPNs Using Routing Table Groups on page 183

Configuring Overlapping VPNs Using Automatic Route Export on page 195

Configure a GRE Tunnel Interface between PE Routers on page 198

Configure a GRE Tunnel Interface between a PE and CE Router on page 205

Configure an ES Tunnel Interface between a PE and CE Router on page 208



Note

The examples in this chapter show only the portions of the configuration that establish VPN functionality. You must also configure other router functionality, including all router interfaces, for a router configuration to work properly.

Configure a Simple Full-Mesh VPN Topology

This example shows how to set up a simple full-mesh service provider VPN configuration, which consists of the following components (see Figure 16):

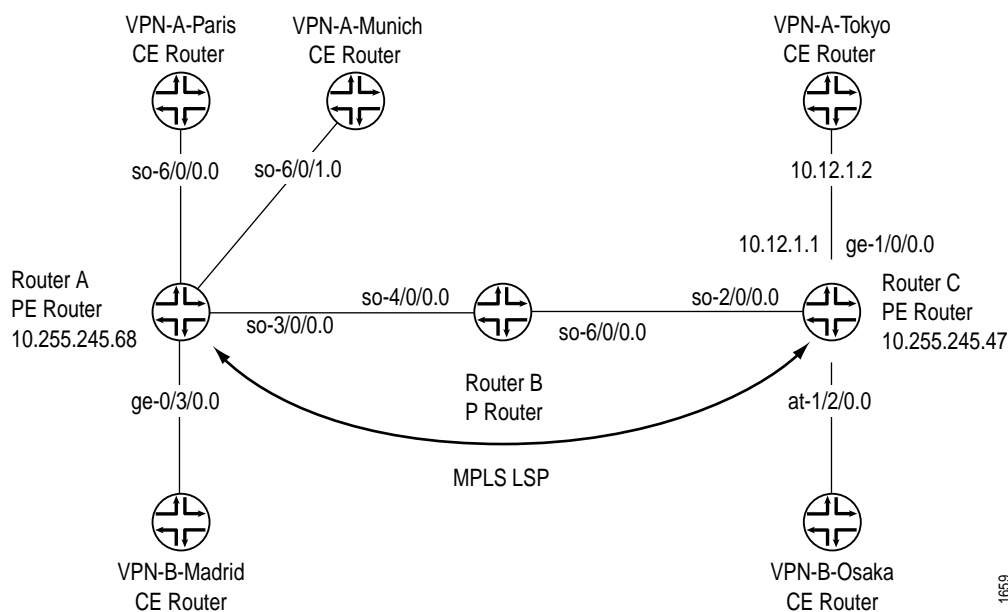
Two separate VPNs (VPN-A and VPN-B)

Two provider edge (PE) routers, both of which service VPN-A and VPN-B

Resource Reservation Protocol (RSVP) as the signaling protocol

One RSVP label-switched path (LSP) that tunnels between the two PE routers through one provider (P) router

Figure 16: Example of a Simple VPN Topology



In this configuration, route distribution in VPN A from the router VPN-A-Paris to the router VPN-A-Tokyo occurs as follows:

1. The customer edge (CE) router VPN-A-Paris announces routes to the PE router Router A.
2. Router A installs the received announced routes into its VPN routing and forwarding (VRF) table, VPN-A.inet.0.
3. Router A creates a Multiprotocol Label Switching (MPLS) label for the interface between it and the router VPN-A-Paris.
4. Router A checks its VRF export policy.
5. Router A converts the Internet Protocol Version 4 (IPv4) routes from VPN-A-Paris into VPN IPv4 format using its route distinguisher and announces these routes to PE Router C over the internal Border Gateway Protocol (IBGP) between the two PE routers.

6. Router C checks its VRF import policy and installs all routes that match the policy into its bgp.l3vpn.0 routing table. (Any routes that do not match are discarded.)
7. Router C checks its VRF import policy and installs all routes that match into its VPN-A.inet.0 routing table. The routes are installed in IPv4 format.
8. Router C announces its routes to the CE router VPN-A-Tokyo, which installs them into its master routing table. (For routers running JUNOS software, the master routing table is inet.0.)
9. Router C uses the LSP between it and Router A to route all packets from router VPN-A-Tokyo that are destined for the router VPN-A-Paris.

The following sections explain how to configure the VPN functionality on the PE and provider routers. The CE routers are not aware of the VPN, so you configure them normally.

Enable an IGP on the PE and Provider Routers on page 129

Enable RSVP and MPLS on the Provider Router on page 130

Configure the MPLS LSP Tunnel between the PE Routers on page 130

Configure IBGP on the PE Routers on page 131

Configure Routing Instances for VPNs on the PE Routers on page 132

Configure VPN Policy on the PE Routers on page 134

The final section in this example, “Simple VPN Configuration Summarized by Router” on page 137, consolidates the statements needed to configure VPN functionality on each of the service provider routers shown in Figure 16.



Note

In this example, a private autonomous system (AS) number is used for the route distinguisher and the route target. This number is used for illustration only. When you are configuring VPNs, you should use an assigned AS number.

Enable an IGP on the PE and Provider Routers

To allow the PE and provider routers to exchange routing information among themselves, you must configure an interior gateway protocol (IGP) on all these routers or you must configure static routes. You configure the IGP on the master instance of the routing protocol process (rpd) (that is, at the [edit protocols] hierarchy level), not within the VPN routing instance (that is, not at the [edit routing-instances] hierarchy level).

You configure the IGP in the standard way. This configuration example does not include this portion of the configuration.

Enable RSVP and MPLS on the Provider Router

On the provider router, Router B, you must configure RSVP and MPLS because this router exists on the MPLS LSP path between the two PE routers, Router A and Router C:

```
[edit]
protocols {
  rsvp {
    interface so-4/0/0.0;
    interface so-6/0/0.0;
  }
  mpls {
    interface so-4/0/0.0;
    interface so-6/0/0.0;
  }
}
```

Configure the MPLS LSP Tunnel between the PE Routers

In this configuration example, RSVP is used for VPN signaling. Therefore, in addition to configuring RSVP, you must enable traffic engineering support in an IGP and you must create an MPLS LSP to tunnel the VPN traffic.

On PE Router A, enable RSVP and configure one end of the MPLS LSP tunnel. In this example, traffic engineering support is enabled for Open Shortest Path First (OSPF). When configuring the MPLS LSP, include interface statements for all interfaces participating in MPLS, including the interfaces to the PE and CE routers. The statements for the interfaces between the PE and CE routers are needed so that the PE router can create an MPLS label for the private interface. In this example, the first interface statement configures MPLS on the interface connected to the LSP, and the remaining three configure MPLS on the interfaces that connect the PE router to the CE routers.

```
[edit]
protocols {
  rsvp {
    interface so-3/0/0.0;
  }
  mpls {
    label-switched-path RouterA-to-RouterC {
      to 10.255.245.47;
    }
    interface so-3/0/0.0;
    interface so-6/0/0.0;
    interface so-6/0/1.0;
    interface ge-0/3/0.0;
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface so-3/0/0.0;
    }
  }
}
```

On PE Router C, enable RSVP and configure the other end of the MPLS LSP tunnel. Again, traffic engineering support is enabled for OSPF, and you configure MPLS on the interfaces to the LSP and the CE routers.

```
[edit]
protocols {
  rsvp {
    interface so-2/0/0.0;
  }
  mpls {
    label-switched-path RouterC-to-RouterA {
      to 10.255.245.68;
    }
    interface so-2/0/0.0;
    interface ge-1/0/0.0;
    interface at-1/2/0.0;
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface so-2/0/0.0;
    }
  }
}
```

Configure IBGP on the PE Routers

On the PE routers, configure an IBGP session with the following properties:

VPN family—To indicate that the IBGP session is for the VPN, include the family inet-vpn statement.

Loopback address—Include the local-address statement, specifying the local PE router's loopback address. The IBGP session for VPNs runs through the loopback address. Note that you must also configure the lo0 interface at the [edit interfaces] hierarchy level. The example does not include this part of the router's configuration.

Neighbor address—Include the neighbor statement, specifying the IP address of the neighboring PE router, which is its loopback (lo0) address.

On PE Router A, configure IBGP as follows:

```
[edit]
protocols {
  bgp {
    group PE-RouterA-to-PE-RouterC {
      type internal;
      local-address 10.255.245.68;
      family inet-vpn {
        unicast;
      }
      neighbor 10.255.245.47;
    }
  }
}
```

On PE Router C, configure IBGP as follows:

```
[edit]
protocols {
  bgp {
    group PE-RouterC-to-PE-RouterA {
      type internal;
      local-address 10.255.245.47;
      family inet-vpn {
        unicast;
      }
      neighbor 10.255.245.68;
    }
  }
}
```

Configure Routing Instances for VPNs on the PE Routers

Both PE routers service VPN-A and VPN-B, so you must configure two routing instances on each router, one for each VPN. For each VPN, you must define the following in the routing instance:

Route distinguisher, which must be unique for each routing instance on the PE router. It is used to distinguish the addresses in one VPN from those in another VPN.

Instance type of vrf, which creates the VRF table on the PE router.

Interfaces connected to the CE routers.

VRF import and export policies, which must be the same on each PE router that services the same VPN. Unless an import policy contains only a then reject statement, it must include reference to a community. Otherwise, when you try to commit the configuration, the commit fails.



Note

In this example, a private AS number is used for the route distinguisher. This number is used for illustration only. When you are configuring VPNs, you should use an assigned AS number.

Routing between the PE and CE routers, which is required for the PE router to distribute VPN-related routes to and from connected CE routers. You can configure a routing protocol—Border Gateway Protocol (BGP), OSPF, or Routing Information Protocol (RIP)—or you can configure static routing.

On PE Router A, configure the following routing instance for VPN-A. In this example, Router A uses static routes to distribute routes to and from the two CE routers to which it is connected.

```
[edit]
routing-instance {
  VPN-A-Paris-Munich {
    instance-type vrf;
    interface so-6/0/0.0;
    interface so-6/0/1.0;
    route-distinguisher 65535:0;
    vrf-import VPN-A-import;
    vrf-export VPN-A-export;
    routing-options {
      static {
        route 172.16.0.0/16 next-hop so-0/0/0.0;
        route 172.17.0.0/16 next-hop so-6/0/1.0;
      }
    }
  }
}
```

On PE Router C, configure the following routing instance for VPN-A. In this example, Router C uses BGP to distribute routes to and from the CE router to which it is connected.

```
[edit]
routing-instance {
  VPN-A-Tokyo {
    instance-type vrf;
    interface ge-1/0/0.0;
    route-distinguisher 65535:1;
    vrf-import VPN-A-import;
    vrf-export VPN-A-export;
    protocols {
      bgp {
        group VPN-A-Site2 {
          peer-as 1;
          neighbor 10.12.1.2;
        }
      }
    }
  }
}
```

On PE Router A, configure the following routing instance for VPN-B. In this example, Router A uses OSPF to distribute routes to and from the CE router to which it is connected.

```
[edit]
routing-instance {
  VPN-B-Madrid {
    instance-type vrf;
    interface ge-0/3/0.0;
    route-distinguisher 65535:2;
    vrf-import VPN-B-import;
    vrf-export VPN-B-export;
    protocols {
      ospf {
        export bgp-to-ospf;
        area 0.0.0.0 {
          interface ge-0/3/0;
        }
      }
    }
  }
}
```

On PE Router C, configure the following routing instance for VPN-B. In this example, Router C uses RIP to distribute routes to and from the CE router to which it is connected.

```
[edit]
routing-instance {
  VPN-B-Osaka {
    instance-type vrf;
    interface at-1/2/0.0;
    route-distinguisher 65535:3;
    vrf-import VPN-B-import;
    vrf-export VPN-B-export;
    protocols {
      rip {
        group PE-C-to-VPN-B {
          export bgp-to-rip;
          neighbor at-1/2/0;
        }
      }
    }
  }
}
```

Configure VPN Policy on the PE Routers

You must configure VPN import and export policies on each of the PE routers so that they install the appropriate routes in their VRF tables, which they use to forward packets within a VPN. For VPN-A, the VRF table is VPN-A.inet.0, and for VPN-B it is VPN-B.inet.0.

In the VPN policy, you also configure VPN target communities.



Note

In this example, a private AS number is used for the route target. This number is used for illustration only. When you are configuring VPNs, you should use an assigned AS number.

On PE Router A, configure the following VPN import and export policies.



Note

The policy qualifiers shown in this example are only those needed for the VPN to function. You can configure additional qualifiers, as needed, to any policies that you configure.

```
[edit]
policy-options {
  policy-statement VPN-A-import {
    term a {
      from {
        protocol bgp;
        community VPN-A;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement VPN-A-export {
    term a {
      from protocol static;
      then {
        community add VPN-A;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  policy-statement VPN-B-import {
    term a {
      from {
        protocol bgp;
        community VPN-B;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement VPN-B-export {
    term a {
      from protocol ospf;
      then {
        community add VPN-B;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  community VPN-A members target:65535:0;
  community VPN-B members target:65535:2;
}
```

On PE Router C, configure the following VPN import and export policies:

```
[edit]
policy-options {
  policy-statement VPN-A-import {
    term a {
      from {
        protocol bgp;
        community VPN-A;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement VPN-A-export {
    term a {
      from protocol bgp;
      then {
        community add VPN-A;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  policy-statement VPN-B-import {
    term a {
      from {
        protocol bgp;
        community VPN-B;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement VPN-B-export {
    term a {
      from protocol rip;
      then {
        community add VPN-B;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  community VPN-A members target:65535:1;
  community VPN-B members target:65535:3;
}
```

To apply the VPN policies on the routers, include the `vrf-export` and `vrf-import` statements when you configure the routing instance. For both VPNs, the VRF import and export policies handle the route distribution across the IBGP session running between the PE routers.

To apply the VPN policies on PE Router A, include the following statements:

```
[edit]
routing-instance {
  VPN-A-Paris-Munich {
    vrf-import VPN-A-import;
    vrf-export VPN-A-export;
  }
  VPN-B-Madrid {
    vrf-import VPN-B-import;
    vrf-export VPN-B-export;
  }
}
```

To apply the VPN policies on PE Router C, include the following statements:

```
[edit]
routing-instance {
  VPN-A-Tokyo {
    vrf-import VPN-A-import;
    vrf-export VPN-A-export;
  }
  VPN-B-Osaka {
    vrf-import VPN-B-import;
    vrf-export VPN-B-export;
  }
}
```

Simple VPN Configuration Summarized by Router

Router A (PE Router)

Routing Instance for VPN-A	<pre>routing-instance { VPN-A-Paris-Munich { instance-type vrf; interface so-6/0/0.0; interface so-6/0/1.0; route-distinguisher 65535:0; vrf-import VPN-A-import; vrf-export VPN-A-export; } }</pre>
Instance Routing Protocol	<pre>routing-options { static { route 172.16.0.0/16 next-hop so-6/0/0.0; route 172.17.0.0/16 next-hop so-6/0/1.0; } }</pre>
Routing Instance for VPN-B	<pre>routing-instance { VPN-B-Madrid { instance-type vrf; interface ge-0/3/0.0; route-distinguisher 65535:2; vrf-import VPN-B-import; vrf-export VPN-B-export; } }</pre>

```

Instance Routing Protocol      protocols {
                                ospf {
                                  area 0.0.0.0 {
                                    interface ge-0/3/0;
                                  }
                                }
                              }
Master Protocol Instance      protocols {

  Enable RSVP                  rsvp {
                                interface so-3/0/0.0;
                              }

  Configure an MPLS LSP        mpls {
                                label-switched-path RouterA-to-RouterC {
                                  to 10.255.245.47;
                                }
                                interface so-3/0/0.0;
                                interface so-6/0/0.0;
                                interface so-6/0/1.0;
                                interface ge-0/3/0.0;
                              }

  Configure IBGP                bgp {
                                group PE-RouterA-to-PE-RouterC {
                                  type internal;
                                  local-address 10.255.245.68;
                                  family inet-vpn {
                                    unicast;
                                  }
                                  neighbor 10.255.245.47;
                                }
                              }

  Configure OSPF for Traffic    ospf {
    Engineering Support          traffic-engineering;
                                area 0.0.0.0 {
                                  interface so-3/0/0.0;
                                }
                              }

  Configure VPN Policy          policy-options {
                                policy-statement VPN-A-import {
                                  term a {
                                    from {
                                      protocol bgp;
                                      community VPN-A;
                                    }
                                    then accept;
                                  }
                                  term b {
                                    then reject;
                                  }
                                }
                              }

```

```

policy-statement VPN-A-export {
  term a {
    from protocol static;
    then {
      community add VPN-A;
      accept;
    }
  }
  term b {
    then reject;
  }
}
policy-statement VPN-B-import {
  term a {
    from {
      protocol bgp;
      community VPN-B;
    }
    then accept;
  }
  term b {
    then reject;
  }
}
policy-statement VPN-B-export {
  term a {
    from protocol ospf;
    then {
      community add VPN-B;
      accept;
    }
  }
  term b {
    then reject;
  }
}
community VPN-A members target:65535:0;
community VPN-B members target:65535:2;
}

```

Router B (Provider Router)

```

Master Protocol Instance protocols {

  Enable RSVP      rsvp {
                    interface so-4/0/0.0;
                    interface so-6/0/0.0;
                  }

  Enable MPLS      mpls {
                    interface so-4/0/0.0;
                    interface so-6/0/0.0;
                  }
}

```

Router C (PE Router)

```

Routing Instance for VPN-A  routing-instance {
                             VPN-A-Tokyo {
                               instance-type vrf;
                               interface ge-1/0/0.0;
                               route-distinguisher 65535:1;
                               vrf-import VPN-A-import;
                               vrf-export VPN-A-export;

Instance Routing Protocol    protocols {
                              bgp {
                                group VPN-A-Site2 {
                                  peer-as 1;
                                  neighbor 10.12.1.2;
                                }
                              }
                              }

Routing Instance for VPN-B  VPN-B-Osaka {
                             instance-type vrf;
                             interface at-1/2/0.0;
                             route-distinguisher 65535:3;
                             vrf-import VPN-B-import;
                             vrf-export VPN-B-export;

Instance Routing Protocol    protocols {
                              rip {
                                group PE-C-to-VPN-B {
                                  neighbor at-1/2/0;
                                }
                              }
                              }

Master Protocol Instance     protocols {

  Enable RSVP                rsvp {
                              interface so-2/0/0.0;
                              }

  Configure an MPLS LSP      mpls {
                              label-switched-path RouterC-to-RouterA {
                                to 10.255.245.68;
                              }
                              interface so-2/0/0.0;
                              interface ge-1/0/0.0;
                              interface at-1/2/0.0;
                              }

  Configure IBGP              bgp {
                              group PE-RouterC-to-PE-RouterA {
                                type internal;
                                local-address 10.255.245.47;
                                family inet-vpn {
                                  unicast;
                                }
                                neighbor 10.255.245.68;
                              }
                              }

```

Configure OSPF for Traffic Engineering Support

```
ospf {
  traffic-engineering;
  area 0.0.0.0 {
    interface so-2/0/0.0;
  }
}
```

Configure VPN Policy

```
policy-options {
  policy-statement VPN-A-import {
    term a {
      from {
        protocol bgp;
        community VPN-A;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement VPN-A-export {
    term a {
      from protocol bgp;
      then {
        community add VPN-A;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  policy-statement VPN-B-import {
    term a {
      from {
        protocol bgp;
        community VPN-B;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement VPN-B-export {
    term a {
      from protocol rip;
      then {
        community add VPN-B;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  community VPN-A members target:65535:1;
  community VPN-B members target:65535:3;
}
```

Configure a Full-Mesh VPN Topology with Route Reflectors

This example is a variation of the full-mesh VPN topology example (described in “Configure a Simple Full-Mesh VPN Topology” on page 128) in which one of the PE routers is a BGP route reflector. In this variation, Router C in Figure 16 on page 128 is a route reflector. The only change to its configuration is that you need to include the cluster statement when configuring the BGP group:

```
[edit protocols]
bgp {
  group PE-RouterC-to-PE-RouterA {
    type internal;
    local-address 10.255.245.47;
    family inet-vpn {
      unicast;
    }
    neighbor 10.255.245.68;
    cluster 4.3.2.1;
  }
}
```

For the complete configuration example of Router C, see “Router C (PE Router)” on page 140.

Configure a Hub-and-Spoke VPN Topology

This example shows how to set up a hub-and-spoke VPN configuration, which consists of the following components (see Figure 17):

One hub PE router (Router D).

One hub CE router connected to the hub PE router. For a hub-and-spoke VPN topology to function properly, there must be two interfaces connecting the hub PE router to the hub CE router, and each interface must have its own VRF table on the PE router:

One interface (here, interface ge-0/0/0.0) is used to announce spoke routes to the hub CE router. The VRF table associated with this interface contains the routes being announced by the spoke PE routers to the hub CE router.

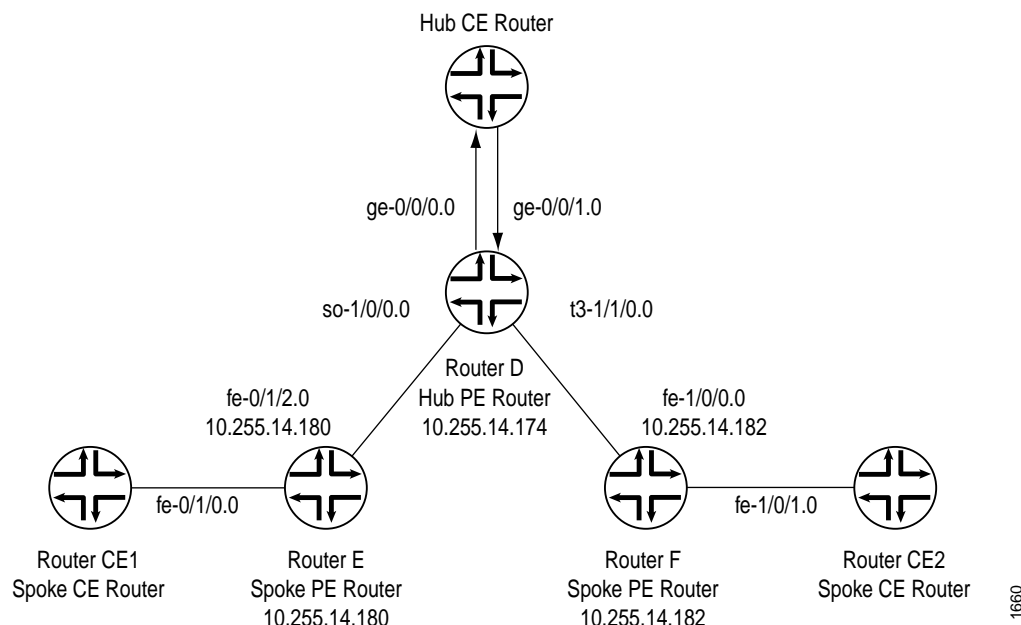
The second interface (here, interface ge-0/0/1.0) is used to receive route announcements from the hub CE that are destined for the hub-and-spoke routers. The VRF table associated with this interface contains the routes announced by the hub CE router to the spoke PE routers.

Two spoke PE routers (Router E and Router F).

Two spoke CE routers (CE1 and CE2), one connected to each spoke PE router.

Label Distribution Protocol (LDP) as the signaling protocol.

Figure 17: Example of a Hub-and-Spoke VPN Topology

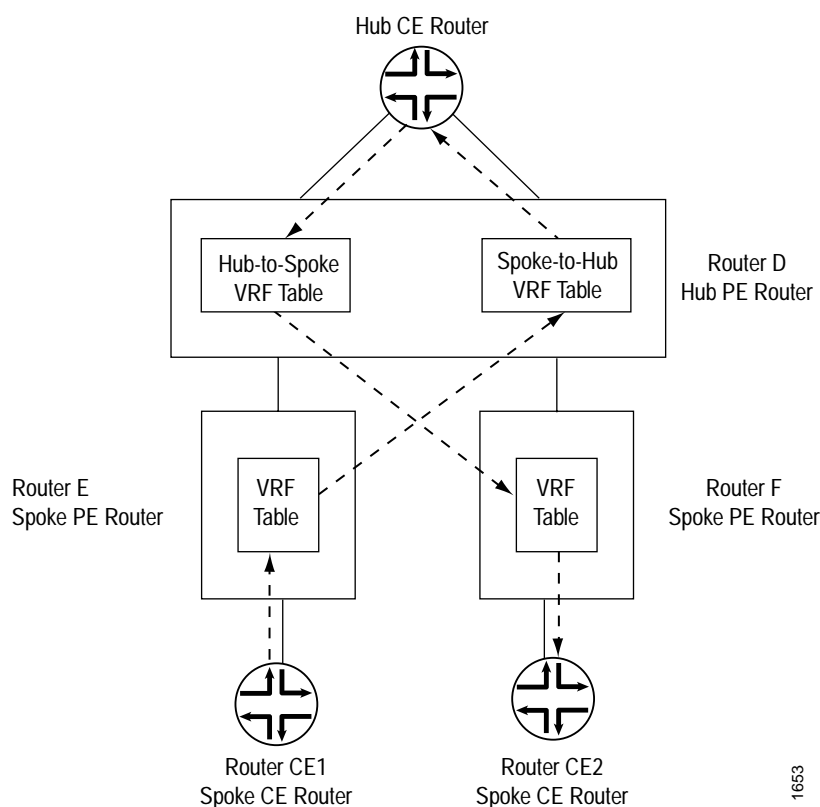


In this configuration, route distribution from spoke CE Router CE1 occurs as follows:

1. Spoke Router CE1 announces its routes to spoke PE Router E.
2. Router E installs the routes from CE1 into its VRF table.
3. After checking its VRF export policy, Router E adds the spoke target community to the routes from Router CE1 that passed the policy and announces them to the hub PE router, Router D.
4. Router D checks the VRF import policy associated with interface ge-0/0/0.0 and places all routes from spoke PE routers that match the policy into its bgp.l3vpn routing table. (Any routes that do not match are discarded.)
5. Router D checks its VRF import policy associated with interface ge-0/0/0.0 and installs all routes that match into its spoke VRF table. The routes are installed with the spoke target community.
6. Router D announces routes to the hub CE over interface ge-0/0/0.
7. The hub CE router announces the routes back to the hub PE Router D over the second interface to the hub router, interface ge-0/0/1.
8. The hub PE router installs the routes learned from the hub CE router into its hub VRF table, which is associated with interface ge-0/0/1.
9. The hub PE router checks the VRF export policy associated with interface ge-0/0/1.0 and announces all routes that match to all spokes after adding the hub target community.

Figure 18 illustrates how routes are distributed from this spoke router to the other spoke CE router, Router CE2. The same path is followed if you issue a traceroute command from Router CE1 to Router CE2.

Figure 18: Route Distribution between Two Spoke Routers



The following sections explain how to configure the VPN functionality for a hub-and-spoke topology on the hub-and-spoke PE routers. The CE routers do not know about the VPN, so you configure them normally.

Enable an IGP on the Hub-and-Spoke PE Routers on page 145

Configure LDP on the Hub-and-Spoke PE Routers on page 145

Configure IBGP on the PE Routers on page 146

Configure Routing Instances for VPNs on the Hub-and-Spoke PE Routers on page 147

Configure VPN Policy on the PE Routers on page 150

The final section in this example, “Hub-and-Spoke VPN Configuration Summarized by Router” on page 153, consolidates the statements needed to configure VPN functionality for each of the service provider routers shown in Figure 17.

Enable an IGP on the Hub-and-Spoke PE Routers

To allow the hub-and-spoke PE routers to exchange routing information, you must configure an IGP on all these routers or you must configure static routes. You configure the IGP on the master instance of the routing protocol process (rpd) (that is, at the [edit protocols] hierarchy level), not within the routing instance (that is, not at the [edit routing-instances] hierarchy level).

You configure the IGP in the standard way. This configuration example does not include this portion of the configuration.

In the route distribution in a hub-and-spoke topology, if the protocol used between the CE and PE routers at the hub site is BGP, the hub CE router announces all routes received from the hub PE router and the spoke routers back to the hub PE router and all the spoke routers. This means that the hub-and-spoke PE routers receive routes that contain their AS number. Normally, when a route contains this information, it indicates that a routing loop has occurred and the router rejects the routes. However, for the VPN configuration to work, the hub PE router and the spoke routers must accept these routes. To enable this, include the loops option when configuring the AS at the [edit routing-options] hierarchy level on the hub PE router and all the spoke routers. For this example configuration, you specify a value of 1. You can specify a number from 0 through 10.

```
[edit routing-options]
autonomous-system as-number loops 1;
```

Configure LDP on the Hub-and-Spoke PE Routers

You must configure LDP on the interfaces between the hub-and-spoke PE routers that participate in the VPN.

On hub PE Router D, configure LDP as follows:

```
[edit protocols]
ldp {
  interface so-1/0/0.0;
  interface t3-1/1/0.0;
}
```

On spoke PE Router E, configure LDP as follows:

```
[edit protocols]
ldp {
  interface fe-0/1/2.0;
}
```

On spoke PE router F, configure LDP as follows:

```
[edit protocols]
ldp {
  interface fe-1/0/0.0;
}
```

Configure IBGP on the PE Routers

On the hub-and-spoke PE routers, configure an IBGP session with the following properties:

VPN family—To indicate that the IBGP session is for the VPN, include the family inet-vpn statement.

Loopback address—Include the local-address statement, specifying the local PE router's loopback address. The IBGP session for VPNs runs through the loopback address. Note that you must also configure the lo0 interface at the [edit interfaces] hierarchy level. The example does not include this part of the router's configuration.

Neighbor address—Include the neighbor statement. On the hub router, specify the IP address of each spoke PE router, and on the spoke router, specify the address of the hub PE router.

For the hub router, you configure an IBGP session with each spoke, and for each spoke router, you configure an IBGP session with the hub. There are no IBGP sessions between the two spoke routers.

On hub Router D, configure IBGP as follows. The first neighbor statement configures an IBGP session to spoke Router E, and the second configures a session to spoke Router F.

```
[edit protocols]
bgp {
  group Hub-to-Spokes {
    type internal;
    local-address 10.255.14.174;
    family inet-vpn {
      unicast;
    }
    neighbor 10.255.14.180;
    neighbor 10.255.14.182;
  }
}
```

On spoke Router E, configure an IBGP session to the hub router as follows:

```
[edit protocols]
bgp {
  group Spoke-E-to-Hub {
    type internal;
    local-address 10.255.14.180;
    neighbor 10.255.14.174 {
      family inet-vpn {
        unicast;
      }
    }
  }
}
```

On spoke Router F, configure an IBGP session to the hub router as follows:

```
[edit protocols]
bgp {
  group Spoke-F-to-Hub {
    type internal;
    local-address 10.255.14.182;
    neighbor 10.255.14.174 {
      family inet-vpn {
        unicast;
      }
    }
  }
}
```

Configure Routing Instances for VPNs on the Hub-and-Spoke PE Routers

For the hub PE router to be able to distinguish between packets going to and coming from the spoke PE routers, you must configure it with two routing instances:

One routing instance (in this example, Spokes-to-Hub-CE) is associated with the interface that carries packets from the hub PE router to the hub CE router (in this example, interface ge-0/0/0.0). Its VRF table contains the routes being announced by the spoke PE routers and the hub PE router to the hub CE router.

The second routing instance (in this example, Hub-CE-to-Spokes) is associated with the interface that carries packets from the hub CE router to the hub PE router (in this example, interface ge-0/0/1.0). Its VRF table contains the routes being announced from the hub CE router to the hub-and-spoke PE routers.

On each spoke router, you must configure one routing instance.

You must define the following in the routing instance:

Route distinguisher, which is used to distinguish the addresses in one VPN from those in another VPN.

Instance type of vrf, which creates the VRF table on the PE router.

Interfaces that are part of the VPN and that connect the PE routers to their CE routers.

VRF import and export policies. Both import policies must include reference to a community. Otherwise, when you try to commit the configuration, the commit fails. (The exception to this is if the import policy contains only a then reject statement.) In the VRF export policy, spoke PE routers attach the spoke target community.

Routing between the PE and CE routers, which is required for the PE router to distribute VPN-related routes to and from connected CE routers. You can configure a routing protocol—BGP, OSPF, or RIP—or you can configure static routing.

For a hub-and-spoke topology, you must configure different policies in each routing instance on the hub CE router. For the routing instance associated with the interface that carries packets from the hub PE router to the hub CE router (in this example, Spokes-to-Hub-CE), the import policy must accept all routes received on the IBGP session between the hub-and-spoke PE routers and the export policy must reject all routes received from the hub CE router. For the routing instance associated with the interfaces that carries packets from the hub CE router to the hub PE router (in this example, Hub-CE-to-Spokes), the import policy must reject all routes received from the spoke PE routers, and the export policy must export to all the spoke routers.

On hub PE Router D, configure the following routing instances. Router D uses OSPF to distribute routes to and from the hub CE router.

```
[edit]
routing-instance {
  Spokes-to-Hub-CE {
    instance-type vrf;
    interface ge-0/0/0.0;
    route-distinguisher 10.255.1.174:65535;
    vrf-import spoke;
    vrf-export null;
    protocols {
      ospf {
        export redistribute-vpn;
        area 0.0.0.0 {
          interface ge-0/0/0;
        }
      }
    }
  }
  Hub-CE-to-Spokes {
    instance-type vrf;
    interface ge-0/0/1.0;
    route-distinguisher 10.255.1.174:65535;
    vrf-import null;
    vrf-export hub;
    protocols {
      ospf {
        export redistribute-vpn;
        area 0.0.0.0 {
          interface ge-0/0/1.0;
        }
      }
    }
  }
}
```

On spoke PE Router E, configure the following routing instances. Router E uses OSPF to distribute routes to and from the spoke CE Router CE1.

```
[edit]
routing-instance {
  Spoke-E-to-Hub {
    instance-type vrf;
    interface fe-0/1/0.0;
    route-distinguisher 10.255.14.80:65535;
    vrf-import hub;
    vrf-export spoke;
    protocols {
      ospf {
        export redistribute-vpn;
        area 0.0.0.0 {
          interface fe-0/1/0.0;
        }
      }
    }
  }
}
```

On spoke PE Router F, configure the following routing instances. Router F uses OSPF to distribute routes to and from the spoke CE Router CE2.

```
[edit]
routing-instance {
  Spoke-F-to-Hub {
    instance-type vrf;
    interface fe-1/0/1.0;
    route-distinguisher 10.255.14.182:65535;
    vrf-import hub;
    vrf-export spoke;
    protocols {
      ospf {
        export redistribute-vpn;
        area 0.0.0.0 {
          interface fe-1/0/1.0;
        }
      }
    }
  }
}
```

Configure VPN Policy on the PE Routers

You must configure VPN import and export policies on each of the hub-and-spoke PE routers so that they install the appropriate routes in the VRF tables, which they use to forward packets within each VPN.

On the spoke routers, you define policies to exchange routes with the hub router.

On the hub router, you define policies to accept routes from the spoke PE routers and distribute them to the hub CE router, and vice versa. The hub PE router has two VRF tables:

Spoke-to-hub VRF table—Handles routes received from spoke routers and announces these routes to the hub CE router. For this VRF table, the import policy must check that the spoke target name is present and that the route was received from the IBGP session between the hub PE and the spoke PE routers. This VRF table must not export any routes, so its export policy should reject everything.

Hub-to-spoke VRF table—Handles routes received from the hub CE router and announces them to the spoke routers. For this VRF table, the export policy must add the hub target community. This VRF table must not import any routes, so its import policy should reject everything.

In the VPN policy, you also configure the VPN target communities.

On hub PE Router D, configure the following policies to apply to the VRF tables:

spoke—Accepts routes received from the IBGP session between it and the spoke PE routers that contain the community target spoke, and rejects all other routes.

hub—Adds the community target hub to all routes received from OSPF (that is, from the session between it and the hub CE router). It rejects all other routes.

null—Rejects all routes.

redistribute-vpn—Redistributes OSPF routes to neighbors within the routing instance.

```
[edit]
policy-options {
  policy-statement spoke {
    term a {
      from {
        protocol bgp;
        community spoke;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
}
```



```

policy-statement hub {
  term a {
    from protocol ospf;
    then {
      community add hub;
      accept;
    }
  }
  term b {
    then reject;
  }
}
policy-statement null {
  then reject;
}
policy-statement redistribute-vpn {
  term a {
    from protocol bgp;
    then accept;
  }
  term b {
    then reject;
  }
}
community hub members target:65535:1;
community spoke members target:65535:2;
}

```

To apply the VRF policies on Router D, include the `vrf-export` and `vrf-import` statements when you configure the routing instances:

```

[edit]
routing-instance {
  Spokes-to-Hub-CE {
    vrf-import spoke;
    vrf-export null;
  }
  Hub-CE-to-Spokes {
    vrf-import null;
    vrf-export hub;
  }
}

```

On spoke PE Router E and Router F, configure the following policies to apply to the VRF tables:

hub—Accepts routes received from the IGBP session between it and the hub PE routers that contain the community target `hub`, and rejects all other routes.

spoke—Adds the community target `spoke` to all routes received from OSPF (that is, from the session between it and the hub CE router) and rejects all other routes.

redistribute-vpn—Redistributes OSPF routes to neighbors within the routing instance.

On spoke PE Router E and Router F, configure the following VPN import and export policies:

```
[edit]
policy-options {
  policy-statement hub {
    term a {
      from {
        protocol bgp;
        community hub;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement spoke {
    term a {
      from protocol ospf;
      then {
        community add spoke;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  policy-statement redistribute-vpn {
    term a {
      from protocol bgp;
      then accept;
    }
    term b {
      then reject;
    }
  }
  community hub members target:65535:1;
  community spoke members target 65535:2;
}
```

To apply the VRF policies on the spoke routers, include the vrf-export and vrf-import statements when you configure the routing instances:

```
[edit]
routing-instance {
  Spoke-E-to-Hub {
    vrf-import hub;
    vrf-export spoke;
  }
}

[edit]
routing-instance {
  Spoke-F-to-Hub {
    vrf-import hub;
    vrf-export spoke;
  }
}
```

Hub-and-Spoke VPN Configuration Summarized by Router

Router D (Hub PE Router)

Routing Instance for Distributing Spoke Routes to Hub CE	<pre> routing-instance { Spokes-to-Hub-CE { instance-type vrf; interface ge-0/0/0.0; route-distinguisher 10.255.1.174:65535; vrf-import spoke; vrf-export null; } } </pre>
Instance Routing Protocol	<pre> protocols { ospf { export redistribute-vpn; area 0.0.0.0 { interface ge-0/0/0; } } } </pre>
Routing Instance for Distributing Hub CE Routes to Spokes	<pre> Hub-CE-to-Spokes { instance-type vrf; interface ge-0/0/1.0; route-distinguisher 10.255.1.174:65535; vrf-import null; vrf-export hub; } </pre>
Instance Routing Protocols	<pre> protocols { ospf { export redistribute-vpn; area 0.0.0.0 { interface ge-0/0/1.0; } } } </pre>
Routing Options (Master Instance)	<pre> routing-options { autonomous-system 1 loops 1; } </pre>
Protocols (Master Instance)	<pre> protocols { </pre>
Enable LDP	<pre> ldp { interface so-1/0/0.0; interface t3-1/1/0.0; } } </pre>

```

Configure IBGP  bgp {
                  group Hub-to-Spokes {
                    type internal;
                    local-address 10.255.14.174;
                    family inet-vpn {
                      unicast;
                    }
                    neighbor 10.255.14.180;
                    neighbor 10.255.14.182;
                  }
                }

Configure VPN Policy  policy-options {
                     policy-statement spoke {
                       term a {
                         from {
                           protocol bgp;
                           community spoke;
                         }
                         then accept;
                       }
                       term b {
                         then reject;
                       }
                     }
                     policy-statement hub {
                       term a {
                         from protocol ospf;
                         then {
                           community add hub;
                           accept;
                         }
                       }
                       term b {
                         then reject;
                       }
                     }
                     policy-statement null {
                       then reject;
                     }
                     policy-statement redistribute-vpn {
                       term a {
                         from protocol bgp;
                         then accept;
                       }
                       term b {
                         then reject;
                       }
                     }
                     community hub members target:65535:1;
                     community spoke members target:65535:2;
                   }

```

Router E (Spoke PE Router)

Routing Instance	<pre> routing-instance { Spoke-E-to-Hub { instance-type vrf; interface fe-0/1/0.0; route-distinguisher 10.255.14.80:65535; vrf-import hub; vrf-export spoke; } } </pre>
Instance Routing Protocol	<pre> protocols { ospf { export redistribute-vpn; area 0.0.0.0 { interface fe-0/1/0.0; } } } </pre>
Routing Options (Master Instance)	<pre> routing-options { autonomous-system 1 loops 1; } </pre>
Protocols (Master Instance)	<pre> protocols { </pre>
Enable LDP	<pre> ldp { interface fe-0/1/2.0; } </pre>
Configure IBGP	<pre> bgp { group Spoke-E-to-Hub { type internal; local-address 10.255.14.180; neighbor 10.255.14.174 { family inet-vpn { unicast; } } } } } </pre>
Configure VPN Policy	<pre> policy-options { policy-statement hub { term a { from { protocol bgp; community hub; } then accept; } term b { then reject; } } } </pre>

```

policy-statement spoke {
  term a {
    from protocol ospf;
    then {
      community add spoke;
      accept;
    }
  }
  term b {
    then reject;
  }
}
policy-statement redistribute-vpn {
  term a {
    from protocol bgp;
    then accept;
  }
  term b {
    then reject;
  }
}
community hub members target:65535:1;
community spoke members target:65535:2;
}

```

Router F (Spoke PE Router)

```

Routing Instance  routing-instance {
                  Spoke-F-to-Hub {
                    instance-type vrf;
                    interface fe-1/0/1.0;
                    route-distinguisher 10.255.14.182:65535;
                    vrf-import hub;
                    vrf-export spoke;

```

```

Instance Routing Protocol  protocols {
                           ospf {
                             export redistribute-vpn;
                             area 0.0.0.0 {
                               interface fe-1/0/1.0;
                             }
                           }
                           }
                           }

```

```

Routing Options (Master Instance) routing-options {
                                   autonomous-system 1 loops 1;
                                   }

```

```

Protocols (Master Instance) protocols {

```

```

Enable LDP      ldp {
                interface fe-1/0/0.0;
                }

```

Configure IBGP

```

bgp {
  group Spoke-F-to-Hub {
    type internal;
    local-address 10.255.14.182;
    neighbor 10.255.14.174 {
      family inet-vpn {
        unicast;
      }
    }
  }
}

```

Configure VPN Policy

```

policy-options {
  policy-statement hub {
    term a {
      from {
        protocol bgp;
        community hub;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement spoke {
    term a {
      from protocol ospf;
      then {
        community add spoke;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  policy-statement redistribute-vpn {
    term a {
      from {
        protocol bgp;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  community hub members target:65535:1;
  community spoke members target:65535:2;
}

```

Configure an LDP-over-RSVP VPN Topology

This example shows how to set up a VPN topology in which LDP packets are tunneled over an RSVP LSP. This configuration consists of the following components (see Figure 19):

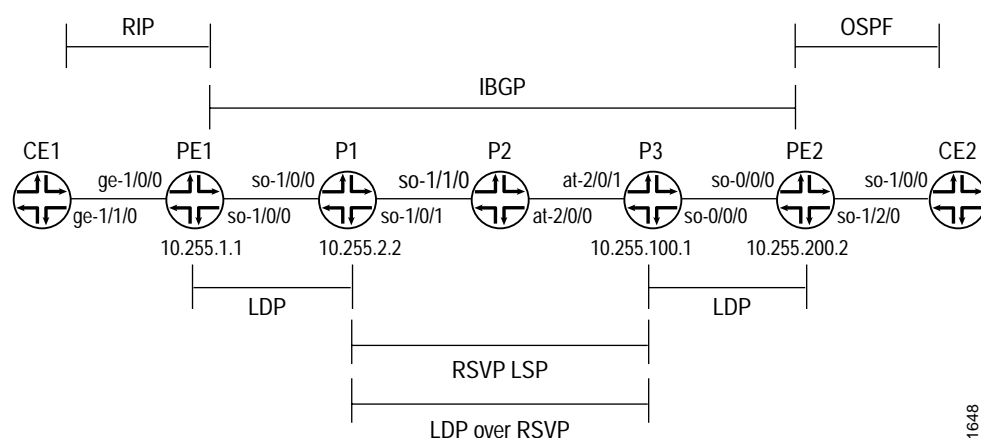
One VPN (VPN-A)

Two PE routers

LDP as the signaling protocol between the PE routers and their adjacent provider routers

An RSVP LSP between two of the provider routers over which LDP is tunneled

Figure 19: Example of an LDP-over-RSVP VPN Topology



The following steps describe how this topology is established and how packets are sent from CE Router CE2 to CE Router CE1:

1. The provider routers P1 and P3 establish RSVP LSPs between each other and install their loopback addresses in their inet.3 routing tables.
2. PE Router PE1 establishes an LDP session with Router P1 over interface so-1/0/0.0.
3. Router P1 establishes an LDP session with Router P3's loopback address, which is reachable using the RSVP LSP.
4. Router P1 sends its label bindings, which include a label to reach Router PE1, to Router P3. These label bindings allow Router P3 to direct LDP packets to Router PE1.
5. Router P3 establishes an LDP session with Router PE2 over interface so0-0/0/0.0 and establishes an LDP session with Router P1's loopback address.
6. Router P3 sends its label bindings, which include a label to reach Router PE2, to Router P1. These label bindings allow Router P1 to direct LDP packets to Router PE2's loopback address.
7. Routers PE1 and PE2 establish IBGP sessions with each other.

8. When Router PE1 announces to Router PE2 routes that it learned from Router CE1, it includes its VPN label. (The PE router creates the VPN label and binds it to the interface between the PE and CE routers.) Similarly, when Router PE2 announces routes that it learned from Router CE2, it sends its VPN label to Router PE1.

When Router PE2 wants to forward a packet to Router CE1, it pushes two labels onto the packet's label stack: first the VPN label that is bound to the interface between Router PE1 and Router CE1, then the LDP label used to reach Router PE1. Then it forwards the packets to Router P3 over interface so-0/0/1.0.

9. When Router P3 receives the packets from Router PE2, it swaps the LDP label that is on top of the stack (according to its LDP database) and also pushes an RSVP label onto the top of the stack so that the packet can now be switched by the RSVP LSP. At this point, there are three labels on the stack: the inner (bottom) label is the VPN label, the middle is the LDP label, and the outer (top) is the RSVP label.
10. Router P2 receives the packet and switches it to Router P1 by swapping the RSVP label. In this topology, because Router P2 is the penultimate-hop router in the LSP, it pops the RSVP label and forwards the packet over interface so-1/1/0.0 to Router P1. At this point, there are two labels on the stack: the inner label is the VPN label and the outer one is the LDP label.
11. When Router P1 receives the packet, it pops the outer label (the LDP label) and forwards the packet to Router PE1 using interface so-1/0/0.0. In this topology, Router PE1 is the egress LDP router, so Router P1 pops the LDP label instead of swapping it with another label. At this point, there is only one label on the stack, the VPN label.
12. When Router PE1 receives the packet, it pops the VPN label and forwards the packet as an IPv4 packet to Router CE1 over interface ge-1/1/0.0.

A similar set of operations occurs for packets sent from Router CE1 that are destined for Router CE2.

The following list explains how, for packets being sent from Router CE2 to Router CE1, the LDP, RSVP, and VPN labels are announced by the various routers. These steps include examples of label values (illustrated in Figure 20).

LDP labels

Router PE1 announces LDP label 3 for itself to Router P1.

Router P1 announces LDP label 100,001 for Router PE1 to Router P3.

Router P3 announces LDP label 100,002 for Router PE1 to Router PE2.

RSVP labels

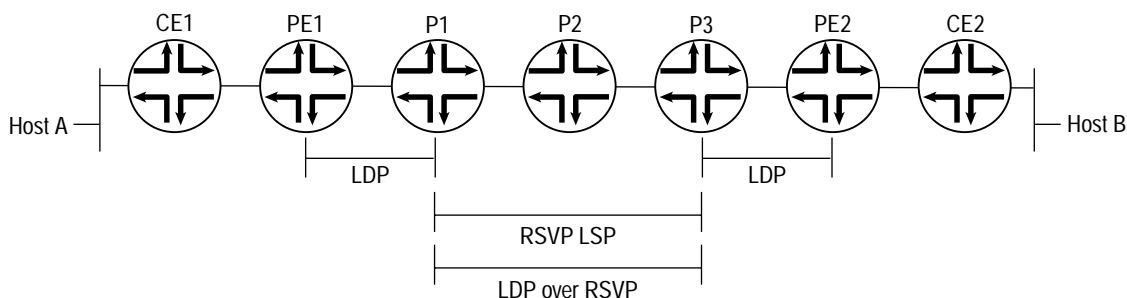
Router P1 announces RSVP label 3 to Router P2.

Router P2 announces RSVP label 100,003 to Router P3.

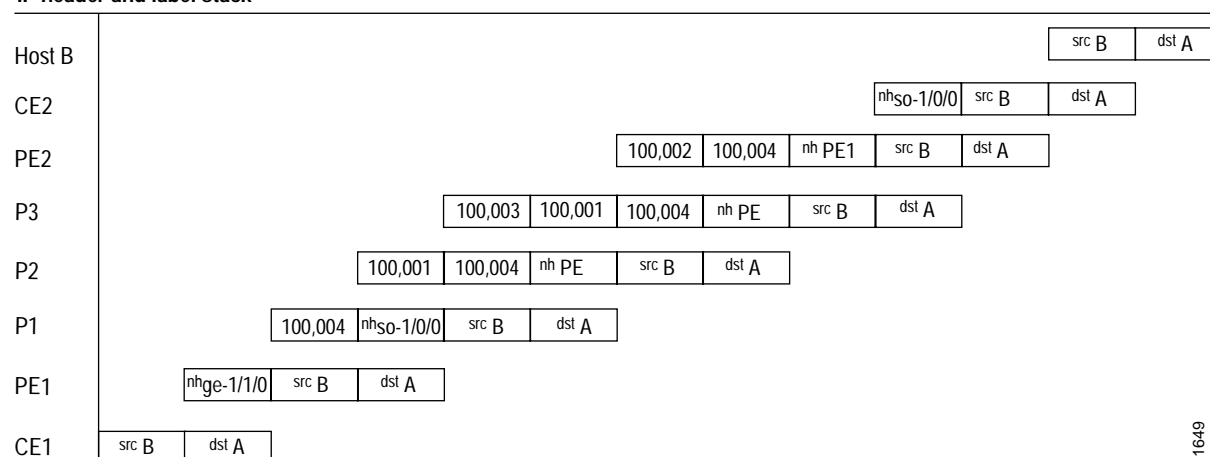
VPN label

Router PE1 announces VPN label 100,004 to Router PE2 for the route from Router CE1 to Router CE2.

Figure 20: Label Pushing and Popping



IP header and label stack



For a packet sent from Host B in Figure 20 to Host A, the packet headers and labels change as follows as the packet travels to its destination:

1. The packet that originates from Host B has a source address of B and a destination address of A in its header.
2. Router CE2 adds to the packet a next-hop of interface so-1/0/0.
3. Router PE2 swaps out the next-hop of interface so-1/0/0 and replaces it with a next-hop of PE1. It also adds two labels for reaching Router PE1, first the VPN label (100,004), then the LDP label (100,002). The VPN label is thus the inner (bottom) label on the stack, and the LDP label is the outer label.
4. Router P3 swaps out the LDP label added by Router PE2 (100,002) and replaces it with its LDP label for reaching Router PE1 (100,001). It also adds the RSVP label for reaching Router P2 (100,003).
5. Router P2 removes the RSVP label (100,003) because it is the penultimate hop in the MPLS LSP.
6. Router P1 removes the LDP label (100,001) because it is the penultimate LDP router. It also swaps out the next-hop of PE1 and replaces it with the next-hop interface, so-1/0/0.

7. Router PE1 removes the VPN label (100,004). It also swaps out the next-hop interface of so-1/0/0 and replaces it with its next-hop interface, ge-1/1/0.
8. Router CE1 removes the next-hop interface of ge-1/1/0, and the packet header now contains just a source address of B and a destination address of A.

The following sections explain how to configure the VPN functionality on the PE and provider routers. The CE routers are not aware of the VPN, so you configure them normally.

Enable an IGP on the PE and Provider Routers on page 161

Enable LDP on the PE and Provider Routers on page 162

Enable RSVP and MPLS on the Provider Router on page 163

Configure the MPLS LSP Tunnel between the Provider Routers on page 163

Configure IBGP on the PE Routers on page 164

Configure Routing Instances for VPNs on the PE Routers on page 165

Configure VPN Policy on the PE Routers on page 166

The final section in this example, “LDP-over-MPLS VPN Configuration Summarized by Router” on page 168, consolidates the statements needed to configure VPN functionality on each of the service provider routers shown in Figure 19.



Note

In this example, a private AS number is used for the route distinguisher and the route target. This number is used for illustration only. When you are configuring VPNs, you should use an assigned AS number.

Enable an IGP on the PE and Provider Routers

To allow the PE and provider routers to exchange routing information among themselves, you must configure an IGP on all these routers or you must configure static routes. You configure the IGP on the master instance of the routing protocol process (rpd) (that is, at the [edit protocols] hierarchy level), not within the VPN routing instance (that is, not at the [edit routing-instances] hierarchy level).

You configure the IGP in the standard way. This configuration example does not include this portion of the configuration.

Enable LDP on the PE and Provider Routers

In this configuration example, the LDP is the signaling protocol between the PE routers. For the VPN to function, you must configure LDP on the two PE routers and on the provider routers that are connected to the PE routers. You need to configure LDP only on the interfaces in the core of the service provider's network; that is, between the PE and provider routers and between the provider routers. You do not need to configure LDP on the interface between the PE and CE routers.

In this configuration example, you configure LDP on the provider routers' loopback interfaces because these are the interfaces on which the MPLS LSP is configured.

On the PE routers, you must also configure family inet when you configure the logical interface.

On Router PE1, configure LDP as follows:

```
[edit protocols]
ldp {
  interface so-1/0/0.0;
}
[edit interfaces]
so-1/0/0 {
  unit 0 {
    family mpls;
  }
}
```

On Router PE2, configure LDP as follows:

```
[edit protocols]
ldp {
  interface so-0/0/0.0;
}
[edit interfaces]
so-0/0/1 {
  unit 0 {
    family mpls;
  }
}
```

On Router P1, configure LDP as follows:

```
[edit protocols]
ldp {
  interface so-1/0/0.0;
  interface lo0;
}
```

On Router P3, configure LDP as follows:

```
[edit protocols]
ldp {
  interface lo0;
  interface so-0/0/0.0;
}
```

On Router P2, although you do not need to configure LDP, you can optionally configure it to provide a fallback LDP path in case the RSVP LSP becomes nonoperational:

```
[edit protocols]
ldp {
  interface so-1/1/0.0;
  interface at-2/0/0.0;
}
```

Enable RSVP and MPLS on the Provider Router

On the provider router, P2, you must configure RSVP and MPLS because this router exists on the MPLS LSP path between the provider Routers P1 and P3:

```
[edit]
protocols {
  rsvp {
    interface so-1/1/0.0;
    interface at-2/0/0.0;
  }
  mpls {
    interface so-1/1/0.0;
    interface at-2/0/0.0;
  }
}
```

Configure the MPLS LSP Tunnel between the Provider Routers

In this configuration example, LDP is tunneled over an RSVP LSP. Therefore, in addition to configuring RSVP, you must enable traffic engineering support in an IGP and you must create an MPLS LSP to tunnel the LDP traffic.

On Router P1, enable RSVP and configure one end of the MPLS LSP tunnel. In this example, traffic engineering support is enabled for OSPF, and you configure MPLS on the interfaces to the LSP and to Router PE1. In the to statement, you specify the loopback address of Router P3.

```
[edit]
protocols {
  rsvp {
    interface so-1/0/1.0;
  }
  mpls {
    label-switched-path P1-to-P3 {
      to 10.255.100.1;
      ldp-tunneling;
    }
    interface so-1/0/0.0;
    interface so-1/0/1.0;
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface so-1/0/0.0;
      interface so-1/0/1.0;
    }
  }
}
```

On Router P3, enable RSVP and configure the other end of the MPLS LSP tunnel. Again, traffic engineering support is enabled for OSPF, and you configure MPLS on the interfaces to the LSP and to Router PE2. In the to statement, you specify the loopback address of Router P1.

```
[edit]
protocols {
  rsvp {
    interface at-2/0/1.0;
  }
  mpls {
    label-switched-path P3-to-P1 {
      to 10.255.2.2;
      ldp-tunneling;
    }
    interface at-2/0/1.0;
    interface so-0/0/0.0;
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface at-2/0/1.0;
      interface so-0/0/0.0;
    }
  }
}
```

Configure IBGP on the PE Routers

On the PE routers, configure an IBGP session with the following properties:

VPN family—To indicate that the IBGP session is for the VPN, include the family inet-vpn statement.

Loopback address—Include the local-address statement, specifying the local PE router's loopback address. The IBGP session for VPNs runs through the loopback address. Note that you must also configure the lo0 interface at the [edit interfaces] hierarchy level. The example does not include this part of the router's configuration.

Neighbor address—Include the neighbor statement, specifying the IP address of the neighboring PE router, which is its loopback (lo0) address.

On Router PE1, configure IBGP as follows:

```
[edit]
protocols {
  bgp {
    group PE1-to-PE2 {
      type internal;
      local-address 10.255.1.1;
      family inet-vpn {
        unicast;
      }
      neighbor 10.255.200.2;
    }
  }
}
```

On Router PE2, configure IBGP as follows:

```
[edit]
protocols {
  bgp {
    group PE2-to-PE1 {
      type internal;
      local-address 10.255.200.2;
      family inet-vpn {
        unicast;
      }
      neighbor 10.255.1.1;
    }
  }
}
```

Configure Routing Instances for VPNs on the PE Routers

Both PE routers service VPN-A, so you must configure one routing instance on each router for the VPN in which you define the following:

Route distinguisher, which must be unique for each routing instance on the PE router. It is used to distinguish the addresses in one VPN from those in another VPN.

Instance type of vrf, which creates the VRF table on the PE router.

Interfaces connected to the CE routers.

VRF import and export policies, which must be the same on each PE router that services the same VPN. Unless the import policy contains only a then reject statement, it must include reference to a community. Otherwise, when you try to commit the configuration, the commit fails.



Note

In this example, a private AS number is used for the route distinguisher. This number is used for illustration only. When you are configuring VPNs, you should use an assigned AS number.

Routing between the PE and CE routers, which is required for the PE router to distribute VPN-related routes to and from connected CE routers. You can configure a routing protocol—BGP, OSPF, or RIP—or you can configure static routing.

On Router PE1, configure the following routing instance for VPN-A. In this example, Router PE1 uses RIP to distribute routes to and from the CE router to which it is connected.

```
[edit]
routing-instance {
  VPN-A {
    instance-type vrf;
    interface ge-1/0/0.0;
    route-distinguisher 65535:0;
    vrf-import VPN-A-import;
    vrf-export VPN-A-export;
    protocols {
      rip {
        group PE1-to-CE1 {
          neighbor ge-1/0/0.0;
        }
      }
    }
  }
}
```

On Router PE2, configure the following routing instance for VPN-A. In this example, Router PE2 uses OSPF to distribute routes to and from the CE router to which it is connected.

```
[edit]
routing-instance {
  VPN-A {
    instance-type vrf;
    interface so-1/2/0.0;
    route-distinguisher 65535:1;
    vrf-import VPN-A-import;
    vrf-export VPN-A-export;
    protocols {
      ospf {
        area 0.0.0.0 {
          interface so-1/2/0.0;
        }
      }
    }
  }
}
```

Configure VPN Policy on the PE Routers

You must configure VPN import and export policies on each of the PE routers so that they install the appropriate routes in their VRF tables, which they use to forward packets within a VPN. For VPN-A, the VRF table is VPN-A.inet.0.

In the VPN policy, you also configure VPN target communities.



Note

In this example, a private AS number is used for the route target. This number is used for illustration only. When you are configuring VPNs, you should use an assigned AS number.

On Router PE1, configure the following VPN import and export policies.



Note

The policy qualifiers shown in this example are only those needed for the VPN to function. You can configure additional qualifiers, as needed, to any policies that you configure.

```
[edit]
policy-options {
  policy-statement VPN-A-import {
    term a {
      from {
        protocol bgp;
        community VPN-A;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement VPN-A-export {
    term a {
      from protocol rip;
      then {
        community add VPN-A;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  community VPN-A members target:65535:00;
}
```

On Router PE2, configure the following VPN import and export policies:

```
[edit]
policy-options {
  policy-statement VPN-A-import {
    term a {
      from {
        protocol bgp;
        community VPN-A;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
}
```

```

policy-statement VPN-A-export {
  term a {
    from protocol ospf;
    then {
      community add VPN-A;
      accept;
    }
  }
  term b {
    then reject;
  }
}
community VPN-A members target:65535:00;
}

```

To apply the VPN policies on the routers, include the vrf-export and vrf-import statements when you configure the routing instance on the PE routers. The VRF import and export policies handle the route distribution across the IBCP session running between the PE routers.

LDP-over-MPLS VPN Configuration Summarized by Router

Router PE1

```

Routing Instance for VPN-A  routing-instance {
                             VPN-A {
                               instance-type vrf;
                               interface ge-1/0/0.0;
                               route-distinguisher 65535:0;
                               vrf-import VPN-A-import;
                               vrf-export VPN-A-export;
                             }
                           }

Instance Routing Protocol    protocols {
                             rip {
                               group PE1-to-CE1 {
                                 neighbor ge-1/0/0.0;
                               }
                             }
                           }

Interfaces                  interfaces {
                             so-1/0/0 {
                               unit 0 {
                                 family mpls;
                               }
                             }
                             ge-1/0/0 {
                               unit 0;
                             }
                           }

Master Protocol Instance    protocols {

Enable LDP                  ldp {
                             interface so-1/0/0.0;
                           }
}

```

```

Enable MPLS      mpls {
                  interface so-1/0/0.0;
                  interface ge-1/0/0.0;
                  }

Configure IBGP    bgp {
                  group PE1-to-PE2 {
                    type internal;
                    local-address 10.255.1.1;
                    family inet-vpn {
                      unicast;
                    }
                    neighbor 10.255.100.1;
                  }
                }

Configure VPN Policy
policy-options {
  policy-statement VPN-A-import {
    term a {
      from {
        protocol bgp;
        community VPN-A;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement VPN-A-export {
    term a {
      from protocol rip;
      then {
        community add VPN-A;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  community VPN-A members target:65535:00;
}

```

Router P1

```

Master Protocol Instance  protocols {

Enable RSVP              rsvp {
                          interface so-1/0/1.0;
                          }

Enable LDP                ldp {
                          interface so-1/0/0.0;
                          interface lo0.0;
                          }

```

```

Enable MPLS      mpls {
                  label-switched-path P1-to-P3 {
                    to 10.255.100.1;
                    ldp-tunneling;
                  }
                  interface so-1/0/0.0;
                  interface so-1/0/1.0;
                }

```

```

Configure OSPF for Traffic Engineering Support  ospf {
                                                  traffic-engineering;
                                                  area 0.0.0.0 {
                                                    interface so-1/0/0.0;
                                                    interface so-1/0/1.0;
                                                  }
                                                  }

```

Router P2

```

Master Protocol Instance  protocols {

  Enable RSVP             rsvp {
                          interface so-1/1/0.0;
                          interface at-2/0/0.0;
                        }

  Enable MPLS             mpls {
                          interface so-1/1/0.0;
                          interface at-2/0/0.0;
                        }
}

```

Router P3

```

Master Protocol Instance  protocols {

  Enable RSVP             rsvp {
                          interface at-2/0/1.0;
                        }

  Enable LDP              ldp {
                          interface so-0/0/0.0;
                          interface lo0.0;
                        }

  Enable MPLS             mpls {
                          label-switched-path P3-to-P1 {
                            to 10.255.2.2;
                            ldp-tunneling;
                          }
                          interface at-2/0/1.0;
                          interface so-0/0/0.0;
                        }
}

```

Configure OSPF for Traffic Engineering Support

```
ospf {
  traffic-engineering;
  area 0.0.0.0 {
    interface at-2/0/1.0;
    interface at-2/0/1.0;
  }
}
```

Router PE2**Routing Instance for VPN-A**

```
routing-instance {
  VPN-A {
    instance-type vrf;
    interface so-1/2/0.0;
    route-distinguisher 65535:1;
    vrf-import VPN-A-import;
    vrf-export VPN-A-export;
  }
}
```

Instance Routing Protocol

```
protocols {
  ospf {
    area 0.0.0.0 {
      interface so-1/2/0.0;
    }
  }
}
```

Interfaces

```
interfaces {
  so-0/0/0 {
    unit 0 {
      family mpls;
    }
  }
  so-1/2/0 {
    unit 0;
  }
}
```

Master Protocol Instance

```
protocols {
```

Enable LDP

```
  ldp {
    interface so-0/0/0.0;
  }
```

Enable MPLS

```
  mpls {
    interface so-0/0/0.0;
    interface so-1/2/0.0;
  }
```

```

Configure IBGP    bgp {
                    group PE2-to-PE1 {
                        type internal;
                        local-address 10.255.200.2;
                        family inet-vpn {
                            unicast;
                        }
                        neighbor 10.255.1.1;
                    }
                }
    
```

```

Configure VPN Policy policy-options {
                    policy-statement VPN-A-import {
                        term a {
                            from {
                                protocol bgp;
                                community VPN-A;
                            }
                            then accept;
                        }
                        term b {
                            then reject;
                        }
                    }
                    policy-statement VPN-A-export {
                        term a {
                            from protocol ospf;
                            then {
                                community add VPN-A;
                                accept;
                            }
                        }
                        term b {
                            then reject;
                        }
                    }
                    community VPN-A members target:65535:01;
                }
    
```

Configure an Application-Based Layer 3 VPN Topology

This example illustrates an application-based mechanism for forwarding traffic into a Layer 3 VPN. Typically, one or more interfaces are associated with, or bound to, a VPN by including them in the configuration of the VPN routing instance. By binding the interface to the VPN, the VPN's VRF table is used to make forwarding decisions for any incoming traffic on that interface. Binding the interface also includes the interface local routes in the VRF, which provides next-hop resolution for VRF routes.

In this example, a firewall filter is used to define which incoming traffic on an interface is forwarded using the standard routing table, inet.0, and which incoming traffic is forwarded using the VRF table. You can expand this example such that incoming traffic on an interface can be redirected to one or more VPNs. For example, you can define a configuration to support a VPN that forwards traffic based on source address, that forwards Hypertext Transfer Protocol (HTTP) traffic, or that forwards only streaming media.

For this configuration to work, the following must be true:

- The interfaces that use filter-based forwarding must not be bound to the VPN.

- Static routing must be used as the means of routing.

- You must define an interface routing table group that is shared among inet.0 and the VRFs to provide local routes to the VRF.

This example consists of two client hosts (Client D and Client E) that are in two different VPNs and that want to send traffic both within the VPN and to the Internet. The paths are defined as follows:

- Client A sends traffic to Client E over VPN A with a return path that also uses VPN A (using the VPN's VRF table).

- Client B sends traffic to Client D over VPN B with a return path that uses standard destination-based routing (using the inet.0 routing table).

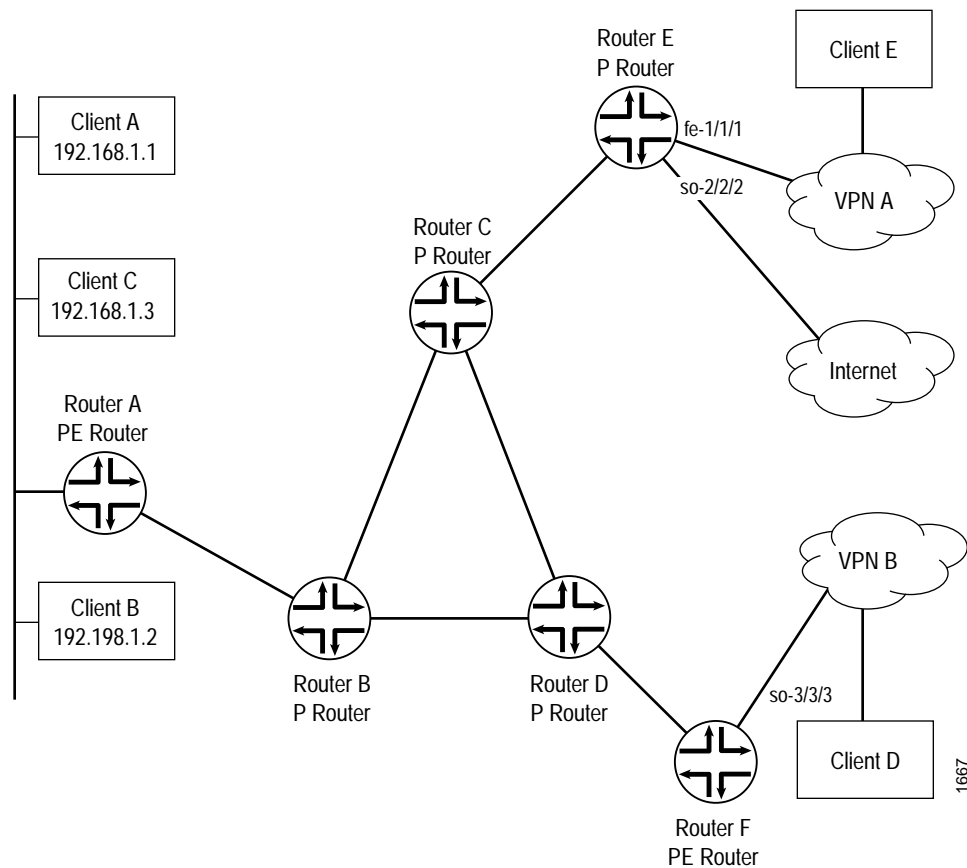
- Clients B and C send traffic to the Internet using standard routing (using the inet.0 routing table), with a return path that also uses standard routing.

This example illustrates that there are a large variety of options in configuring an application-based Layer 3 VPN topology. This flexibility has application in many network implementations requiring specific traffic to be forwarded in a constrained routing environment.

This configuration example shows only the portions of the configuration for the filter-based forwarding, routing instances, and policy. It does not illustrate how to configure a Layer 3 VPN.

Figure 21 illustrates the configuration used in this example.

Figure 21: Application-Based Layer 3 VPN Example Configuration



Configuration on Router A

On Router A, you configure the interface to Clients A, B, and C. The configuration evaluates incoming traffic to determine whether it is to be forwarded using the VPN or using standard destination-based routing.

First, you apply an inbound filter and configure the interface.

```
[edit]
interfaces {
  fe-1/1/0 {
    unit 0 {
      family inet {
        filter {
          input fbv-vrf;
        }
        address 192.168.1.1/24;
      }
    }
  }
}
```


Because the interfaces that use filter-based forwarding must not be bound to a VPN, you must configure an alternate method to provide next-hop routes to the VRF table. You do this by defining an interface routing table group and sharing this group among all the routing tables. To provide a route back to the clients for normal inet.0 routing, you define a static route to include in inet.0 and redistribute the static route into BGP.

```
[edit]
routing-options {
  interface-routes {
    rib-group inet if-rib;
  }
  static {
    route 192.168.1.0/24 next-hop fe-1/1/0.0
  }
  rib-groups {
    if-rib {
      import-rib [ inet.0 vpn-A.inet.0 vpn-B.inet.0 ];
    }
  }
}
```

You apply the following filter to incoming traffic on interface fe-1/1/0.0. The first term matches traffic from Client A and forwards it to the routing instance for VPN A. The second term matches traffic from Client B that is destined for Client D and forwards it to the routing instance for VPN B. The third term matches all other traffic, which is forwarded normally using destination-based forwarding according to the routes in inet.0.

```
[edit firewall family family-name]
filter fbf-vrf {
  term vpnA {
    from {
      source-address {
        192.168.1.1/32;
      }
    }
    then {
      routing-instance vpn-A;
    }
  }
  term vpnB {
    from {
      source-address {
        192.168.1.2/32;
      }
      destination-address {
        192.168.3.0/24;
      }
    }
    then routing-instance vpn-B;
  }
  term internet {
    then accept;
  }
}
```

You then configure the routing instances for VPN A and VPN B. Notice that these statements include all the required statements to define a Layer 3 VPN except for the interface statement.

```
[edit]
routing-instances {
  vpn-A {
    instance-type vrf;
    route-distinguisher 172.21.10.63:100;
    vrf-import vpn-A-import;
    vrf-export vpn-A-export;
    routing-options {
      static {
        route 192.168.1.0/24 next-hop fe-1/1/0.0;
      }
    }
  }
  vpn-B {
    instance-type vrf;
    route-distinguisher 172.21.10.63:200;
    vrf-import vpn-B-import;
    vrf-export vpn-B-export;
    routing-options {
      static {
        route 192.168.1.0/24 next-hop fe-1/1/0.0;
      }
    }
  }
}
```

Configuration on Router E

On Router E, you configure a default route to reach the Internet. You should inject this route into the local IGP mesh to provide an exit point from the network.

```
[edit]
routing-options {
  static {
    route 0.0.0.0/0 next-hop so-2/2/2.0 discard
  }
}
```

You configure the interface to Client E so that all incoming traffic on interface fe-1/1/1.0 that matches the VPN policy is forwarded over VPN A:

```
[edit]
routing-instances {
  vpn-A {
    interface fe-1/1/1.0
    instance-type vrf;
    route-distinguisher 172.21.10.62:100;
    vrf-import vpn-A-import;
    vrf-export vpn-A-export;
    routing-options {
      static {
        route 192.168.2.0/24 next-hop fe-1/1/1.0;
      }
    }
  }
}
```

Configuration for Router F

Again, because the interfaces that use filter-based forwarding must not be bound to a VPN, you configure an alternate method to provide next-hop routes to the VRF table by defining an interface routing table group and sharing this group among all the routing tables. To provide a route back to the clients for normal inet.0 routing, you define a static route to include in inet.0 and redistribute the static route into BGP.

```
[edit]
routing-options {
  interface-routes {
    rib-group inet if-rib;
  }
  rib-groups {
    if-rib {
      import-rib [ inet.0 vpn-B.inet.0];
    }
  }
}
```

To direct traffic from VPN B to Client D, you configure the routing instance for VPN B on Router F. All incoming traffic from Client D on interface so-3/3/3.0 is forwarded normally using the destination address based on the routes in inet.0.

```
[edit]
routing-instances {
  vpn-B {
    instance-type vrf;
    route-distinguisher 172.21.10.64:200;
    vrf-import vpn-B-import;
    vrf-export vpn-B-export;
    routing-options {
      static {
        route 192.168.3.0/24 next-hop so-3/3/3.0;
      }
    }
  }
}
```

Configure an OSPF Domain ID for a Layer 3 VPN

This example illustrates how to configure an OSPF domain ID for a VPN using OSPF as the routing protocol between the PE and CE routers. Routes from an OSPF domain need an OSPF domain ID when they are distributed in BGP as VPN-IPv4 routes in VPNs with multiple OSPF domains. In a VPN connecting multiple OSPF domains, the routes from one domain might overlap with the routes of another.

Configuring a unique OSPF domain ID for each domain ensures that the routes for each domain remain separate. If a domain ID is not configured, the default value is 0.0.0.0. In addition, if the remote PE router does not advertise a domain ID in the VPN-IPv4 routes, the local PE router assumes the domain ID matches the remote PE routers, and an OSPF Type 3 LSA is issued for the routes. Each VRF table in a PE router associated with an OSPF instance is configured with the same OSPF domain ID.

Whether a route is redistributed and advertised as a Type 3 LSA or as a Type 5 LSA depends on the following:

If the receiving PE router sees a Type 3 route with a matching domain ID, the route is redistributed and advertised as a Type 3 LSA.

If the receiving PE router sees a Type 5 route with a matching domain ID, the route is redistributed and advertised as a Type 5 LSA.

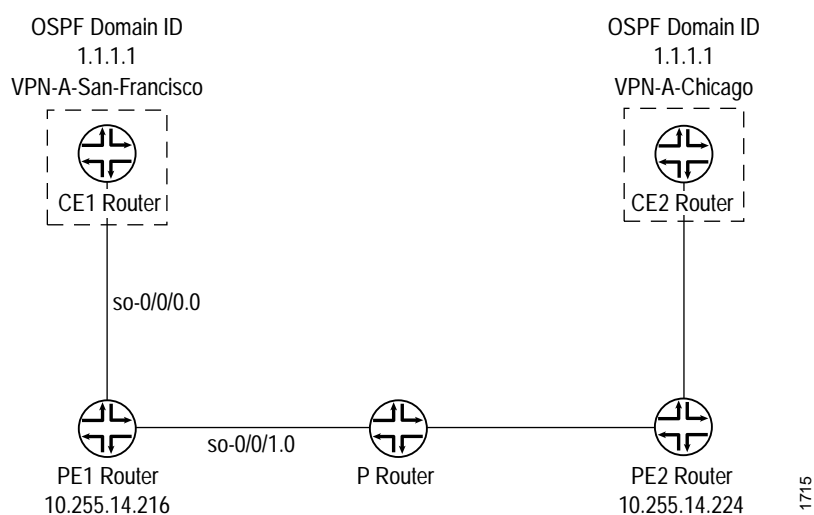
If the receiving PE router sees a Type 3 route without a domain ID, the route is redistributed and advertised as a Type 3 LSA.

If the receiving PE router sees a Type 3 route with a non-matching domain ID, the route is redistributed and advertised as a Type 5 LSA.

If the receiving PE router sees a Type 5 route with a non-matching domain ID, the route is redistributed and advertised as a Type 5 LSA.

Figure 22 shows this example's configuration topology. Only the configuration for Router PE1 is provided. The configuration for Router PE2 can be similar to the configuration for Router PE1. There are no special configuration requirements for the CE routers.

Figure 22: Example of a Configuration Using an OSPF Domain ID



Configure Interfaces on Router PE1

You need to configure two interfaces for Router PE1—the so-0/0/0 interface for traffic to Router CE1 (San Francisco) and the so-0/0/1 interface for traffic to a provider (P) router in the service provider's network.

Configure the interfaces for Router PE1:

```
[edit]
interfaces {
  so-0/0/0 {
    unit 0 {
      family inet {
        address 10.19.1.2/30;
      }
    }
  }
  so-0/0/1 {
    unit 0 {
      family inet {
        address 10.19.2.1/30;
      }
      family mpls;
    }
  }
}
```

Configure Routing Options on Router PE1

At the [edit routing-options] hierarchy level, you need to configure the router-id and autonomous-system statements. The router-id statement identifies Router PE1.

Configure the routing options for Router PE1:

```
[edit]
routing-options {
  router-id 10.255.14.216;
  autonomous-system 69;
}
```

Configure Protocols on Router PE1

On Router PE1, you need to configure MPLS, BGP, OSPF, and LDP at the [edit protocols] hierarchy level:

```
[edit]
protocols {
  mpls {
    interface so-0/0/0.0;
  }
  bgp {
    group San-Francisco-Chicago {
      type internal;
      preference 10;
      local-address 10.255.14.216;
      family inet-vpn {
        unicast;
      }
      neighbor 10.255.14.224;
    }
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface so-0/0/1.0;
    }
  }
  ldp {
    interface so-0/0/1.0;
  }
}
```

Configure Policy Options on Router PE1

On Router PE1, you need to configure policies at the [edit policy-options] hierarchy level. These policies ensure that the CE routers in the Layer 3 VPN exchange routing information. In this example, Router CE1 in San Francisco exchanges routing information with Router CE2 in Chicago.

Configure the policy options on the PE1 router:

```
[edit]
policy-options {
  policy-statement vpn-import-VPN-A {
    term term1 {
      from {
        protocol bgp;
        community import-target-VPN-A;
      }
      then accept;
    }
    term term2 {
      then reject;
    }
  }
}
```

```

policy-statement vpn-export-VPN-A {
  term term1 {
    from protocol ospf;
    then {
      community add export-target-VPN-A;
      accept;
    }
  }
  term term2 {
    then reject;
  }
}
community export-target-VPN-B members [ target:10.255.14.216:11 domain-id:1.1.1.1:0 ];
community import-target-VPN-B members target:10.255.14.224:31;
}

```

Configure the Routing Instance on Router PE1

You need to configure a Layer 3 VPN routing instance on Router PE1. To indicate that the routing instance is for a Layer 3 VPN, add the instance-type vrf statement at the [edit routing-instance *routing-instance-name*] hierarchy level.

The domain-id statement is configured at the [edit routing-instances routing-options protocols ospf] hierarchy level. As shown in Figure 22 on page 178, the routing instance on Router PE2 must share the same domain ID as the corresponding routing instance on Router PE1 so that routes from Router CE1 to Router CE2 and vice versa are distributed as Type 3 LSAs. If you configure different OSPF domain IDs in the routing instances for Router PE1 and Router PE2, the routes from each CE router will be distributed as Type 5 LSAs.

Configure the routing instance on Router PE1:

```

[edit]
routing-instances {
  VPN-A-San-Francisco-Chicago {
    instance-type vrf;
    interface so-0/0/0.0;
    route-distinguisher 10.255.14.216:11;
    vrf-import vpn-import-VPN-A;
    vrf-export vpn-export-VPN-A;
    routing-options {
      router-id 10.255.14.216;
      autonomous-system 69;
    }
    protocols {
      ospf {
        domain-id 1.1.1.1;
        export vpn-import-VPN-A;
        area 0.0.0.0 {
          interface so-0/0/0.0;
        }
      }
    }
  }
}

```

Configuration Summary for Router PE1

```

Configure Interfaces interfaces {
    so-0/0/0 {
        unit 0 {
            family inet {
                address 10.19.1.2/30;
            }
        }
    }
    so-0/0/1 {
        unit 0 {
            family inet {
                address 10.19.2.1/30;
            }
            family mpls;
        }
    }
}

Configure Routing Options routing-options {
    router-id 10.255.14.216;
    autonomous-system 69;
}

Configure Protocols protocols {
    mpls {
        interface so-0/0/0.0;
    }
    bgp {
        group San-Francisco-Chicago {
            type internal;
            preference 10;
            local-address 10.255.14.216;
            family inet-vpn {
                unicast;
            }
            neighbor 10.255.14.224;
        }
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface so-0/0/1.0;
        }
    }
    ldp {
        interface so-0/0/1.0;
    }
}

Configure VPN Policy policy-options {
    policy-statement vpn-import-VPN-A {
        term term1 {
            from {
                protocol bgp;
                community import-target-VPN-A;
            }
            then accept;
        }
    }
}

```



```

        term term2 {
            then reject;
        }
    }
    policy-statement vpn-export-VPN-A {
        term term1 {
            from protocol ospf;
            then {
                community add export-target-VPN-A;
                accept;
            }
        }
        term term2 {
            then reject;
        }
    }
    community export-target-VPN-B members [ target:10.255.14.216:11 domain-id:1.1.1.1:0 ];
    community import-target-VPN-B members target:10.255.14.224:31;
}

```

Routing Instance for Layer 3 VPN

```

routing-instances {
    VPN-A-San-Francisco-Chicago {
        instance-type vrf;
        interface so-0/0/0.0;
        route-distinguisher 10.255.14.216:11;
        vrf-import vpn-import-VPN-A;
        vrf-export vpn-export-VPN-A;
        routing-options {
            router-id 10.255.14.216;
            autonomous-system 69;
        }
        protocols {
            ospf {
                domain-id 1.1.1.1;
                export vpn-import-VPN-A;
                area 0.0.0.0 {
                    interface so-0/0/0.0;
                }
            }
        }
    }
}

```

Configure Overlapping VPNs Using Routing Table Groups

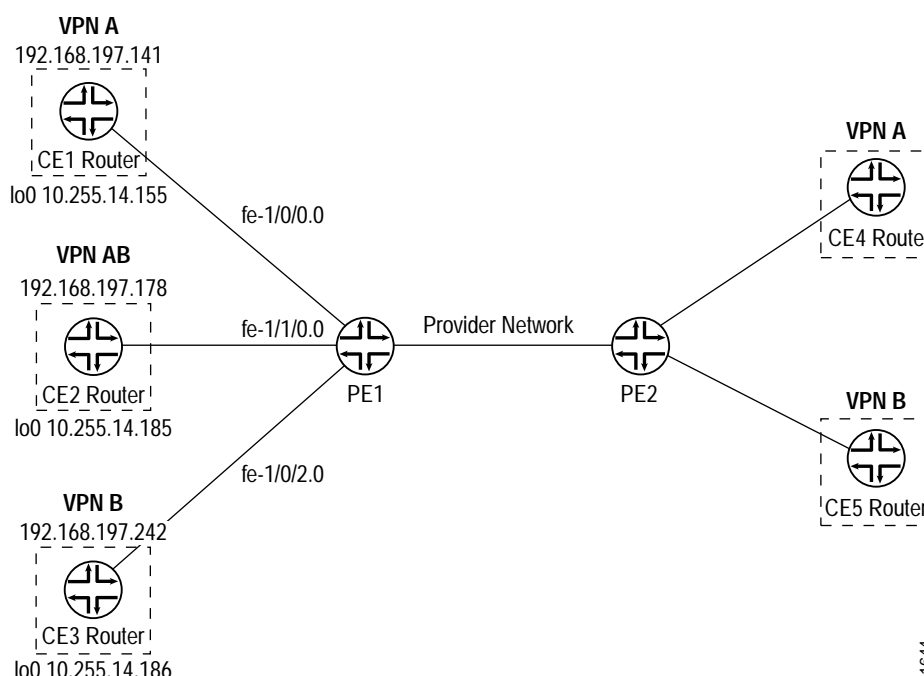
In Layer 3 VPNs, a CE router is often a member of more than one VPN. This example illustrates how to configure PE routers that support CE routers that support multiple VPNs. Support for this type of configuration uses a JUNOS software feature called routing table groups (sometimes also called routing information base [RIB] groups), which allows a route to be installed into several routing tables. A routing table group is a list of routing tables into which the protocol should install its routes.

You define routing table groups at the [edit routing-options] hierarchy level for the default instance. You cannot configure routing table groups at the [routing-instances routing-options] hierarchy level; doing so results in a commit error.

After you define a routing table group, it can be used by multiple protocols. You can also apply routing table groups to static routing. The configuration examples in this section include both types of configurations.

Figure 23 illustrates the topology for the configuration example in this section. The configurations in this section illustrate local connectivity between CE routers connected to the same PE router. If Router PE1 were connected only to Router CE2 (VPN AB), there would be no need for any extra configuration. The configuration statements in the sections that follow enable VPN AB Router CE2 to communicate with VPN A Router CE1 and VPN B Router CE3 that are directly connected to the Router PE1. VPN routes that originate from the remote PE routers (the PE2 router in this case) are placed in a global Layer 3 VPN routing table (bgp.l3vpn.inet.0) and routes with appropriate route targets are imported into the routing tables as dictated by the VRF import policy configuration. The goal is to be able to choose routes from individual VPN routing tables that are locally populated.

Figure 23: Example of an Overlapping VPN Topology



The following sections explain several ways to configure overlapping VPNs. For all the examples that follow, you need to configure routing table groups as described in "Configure Routing Table Groups" on page 185.

The following sections illustrate different scenarios for configuring overlapping VPNs, depending on the routing protocol used between the PE and CE routers. For all of these examples, you need to configure routing table groups.

Configure Static Routes between the PE and CE Routers on page 185

Configure BGP between the PE and CE Routers on page 190

Configure OSPF between the PE and CE Routers on page 192

Configure Static, BGP, and OSPF Routes between the PE and CE Routers on page 193

Router PE1 is where all the filtering and configuration modification takes place. Therefore only VPN configurations for PE1 are shown. The CE routers do not know the VPN exists, so you can configure them normally.

Configure Routing Table Groups

In this example, routing table groups are common in the four configuration scenarios. The routing table groups are used to install routes (including interface, static, OSPF, and BGP routes) into several routing tables for the default and other instances. In the routing table group definition, the first routing table is called the primary routing table. (Normally, the primary routing table is the table into which the route would be installed if you did not configure routing table groups. The other routing tables are called secondary routing tables.)

The routing table groups in this configuration install routes as follows:

vpna-vpnab installs routes into routing tables VPN-A.inet.0 and VPN-AB.inet.0.

vpnb-vpnab installs routes into routing tables VPN-B.inet.0 and VPN-AB.inet.0.

vpnab-vpna_and_vpnab installs routes into routing tables VPN-AB.inet.0, VPN-A.inet.0, and VPN-B.inet.0.

Configure the routing table groups as follows:

```
[edit]
routing-options {
  rib-groups {
    vpna-vpnab {
      import-rib [ VPN-A.inet.0 VPN-AB.inet.0 ];
    }
    vpb-vpnab {
      import-rib [ VPN-B.inet.0 VPN-AB.inet.0 ];
    }
    vpnab-vpna_and_vpnab {
      import-rib [ VPN-AB.inet.0 VPN-A.inet.0 VPN-B.inet.0 ];
    }
  }
}
```

Configure Static Routes between the PE and CE Routers

To configure static routing between the PE1 router and the CE1, CE2, and CE3 routers, you must configure routing instances for VPN A, VPN B, and VPN AB (you configure static routing under each instance):

Configure the Routing Instance for VPN A on page 186

Configure the Routing Instance for VPN AB on page 186

Configure the Routing Instance for VPN B on page 187

Configure VPN Policy on page 188

Configure the Routing Instance for VPN A

On Router PE1, configure VPN A as follows:

```
[edit]
routing-instances {
  VPN-A {
    instance-type vrf;
    interface fe-1/0/0.0;
    route-distinguisher 10.255.14.175:3;
    vrf-import vpn-a-import;
    vrf-export vpn-a-export;
    routing-options {
      interface-routes {
        rib-group inet vpn-a-vpnab;
      }
      static {
        route 10.255.14.155/32 next-hop 192.168.197.141;
        route 10.255.14.185/32 next-hop 192.168.197.178;
      }
    }
  }
}
```

The interface-routes statement installs VPN A's interface routes into the routing tables defined in the routing table group vpn-a-vpnab.

The static statement configures the static routes that are installed in the VPN-A.inet.0 routing table. The first static route is for Router CE1 (VPN A) and the second is for Router CE2 (in VPN AB).

Note that next-hop 192.168.197.178 is not in VPN A. Route 10.255.14.185/32 cannot be installed in VPN-A.inet.0 unless interface routes from routing instance VPN AB are installed in this routing table. Including the interface-routes statements in the VPN AB configuration provides this next hop. Similarly, including the interface-routes statement in the VPN AB configuration installs 192.168.197.141 into VPN-AB.inet.0.

Configure the Routing Instance for VPN AB

On Router PE1, configure VPN AB as follows:

```
[edit]
routing-instances {
  VPN-AB {
    instance-type vrf;
    interface fe-1/1/0.0;
    route-distinguisher 10.255.14.175:9;
    vrf-import vpnab-import;
    vrf-export vpnab-export;
    routing-options {
      interface-routes {
        rib-group vpnab-vpna_and_vpnab;
      }
    }
  }
}
```

```

static {
    route 10.255.14.185/32 next-hop 192.168.197.178;
    route 10.255.14.155/32 next-hop 192.168.197.141;
    route 10.255.14.186/32 next-hop 192.168.197.242;
}
}
}
}

```

In this configuration, the following static routes are installed in the VPN-AB.inet.0 routing table:

10.255.14.185/32 is for Router CE2 (in VPN AB)

10.255.14.155/32 is for Router CE1 (in VPN A)

10.255.14.186/32 is for Router CE3 (in VPN B)

192.168.197.141 and 192.168.197.242 do not belong to VPN AB. Routes 10.255.14.155/32 and 10.255.14.186/32 cannot be installed in VPN-AB.inet.0 unless interface routes from VPN A and VPN B are installed in this routing table. The interface route configurations in VPN A and VPN B routing instances provide these next hops.

Configure the Routing Instance for VPN B

On Router PE1, configure VPN B as follows:

```

[edit]
routing instances {
    VPN-B {
        instance-type vrf;
        interface fe-1/0/2.0;
        route-distinguisher 10.255.14.175:10;
        vrf-import vpnb-import;
        vrf-export vpnb-export;
        routing-options {
            interface-routes {
                rib-group inet vpnb-vpnab;
            }
            static {
                route 10.255.14.186/32 next-hop 192.168.197.242;
                route 10.255.14.185/32 next-hop 192.168.197.178;
            }
        }
    }
}

```

When you configure the routing instance for VPN B, these static routes are placed in VPNB.inet.0:

10.255.14.186/32 is for Router CE3 (in VPN B)

10.255.14.185/32 is for Router CE2 (in VPN AB)

192.168.197.178 does not belong to VPN B. Route 10.255.14.185/32 cannot be installed in VPN-B.inet.0 unless interface routes from VPN AB are installed in this routing table. The interface route configuration in VPN AB provides this next hop.

Configure VPN Policy

The vrf-import and vrf-export policy statements that you configure for overlapping VPNs are the same as policy statements for regular VPNs, except that you include the from interface statement in each VRF export policy. This statement forces each VPN to announce only those routes that originated from that VPN. For example, VPN A has routes that originated in VPN A and VPN AB. If you do not include the from interface statement, VPN A announces its own routes as well as VPN AB's routes, so the remote PE router receives multiple announcements for the same routes. Including the from interface statement restricts each VPN to announcing only the routes it originated and allows you to filter out the routes imported from other routing tables for local connectivity.

In this configuration example, the vpnab-import policy accepts routes from VPN A, VPN B, and VPN AB. The vpna-export policy only exports routes that originate in VPN A. Similarly, the vpnb-export and vpnab-export policies only export routes that originate within the respective VPNs.

On Router PE1, configure the following VPN import and export policies:

```
[edit]
policy-options {
  policy-statement vpna-import {
    term a {
      from {
        protocol bgp;
        community VPNA-comm;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement vpnb-import {
    term a {
      from {
        protocol bgp;
        community VPNB-comm;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement vpnab-import {
    term a {
      from {
        protocol bgp;
        community [ VPNA-comm VPNB-comm ];
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
}
```

```

policy-statement vpna-export {
  term a {
    from {
      protocol static;
      interface fe-1/0/0.0;
    }
    then {
      community add VPNA-comm;
      accept;
    }
  }
  term b {
    then reject;
  }
}
policy-statement vpnb-export {
  term a {
    from {
      protocol static;
      interface fe-1/0/2.0;
    }
    then {
      community add VPNB-comm;
      accept;
    }
  }
  term b {
    then reject;
  }
}
policy-statement vpnab-export {
  term a {
    from {
      protocol static;
      interface fe-1/1/0.0;
    }
    then {
      community add VPNB-comm;
      community add VPNA-comm;
      accept;
    }
  }
  term b {
    then reject;
  }
}
community VPNA-comm members target:69:1;
community VPNB-comm members target:69:2;
}

```

On Router PE1, apply the VPN import and export policies as follows:

```

[edit]
routing-instances {
  VPN-A {
    instance-type vrf;
    interface fe-1/0/0.0;
    route-distinguisher 10.255.14.175:3;
    vrf-import vpna-import;
    vrf-export vpna-export;
  }
}

```

```

routing-options {
  static {
    rib-group vpnab-vpnab;
    route 10.255.14.155/32 next-hop 192.168.197.141;
    route 10.255.14.185/32 next-hop 192.168.197.178;
  }
}
VPN-AB {
  instance-type vrf;
  interface fe-1/1/0.0;
  route-distinguisher 10.255.14.175:9;
  vrf-import vpnab-import;
  vrf-export vpnab-export;
  routing-options {
    static {
      rib-group vpnab-vpna_and_vpnab;
      route 10.255.14.185/32 next-hop 192.168.197.178;
    }
  }
}
VPN-B {
  instance-type vrf;
  interface fe-1/0/2.0;
  route-distinguisher 10.255.14.175:10;
  vrf-import vpnb-import;
  vrf-export vpnb-export;
  routing-options {
    static {
      rib-group vpnb-vpnab;
      route 10.255.14.186/32 next-hop 192.168.197.242;
    }
  }
}

```

For VPN A, include the routing-options statement at the [edit routing-instances *routing-instance-name*] hierarchy level to install the static routes directly into the routing tables defined in the routing table group vpnab-vpnab. For VPN AB, the configuration installs the static route directly into the routing tables defined in the routing table group vpnab-vpna and vpnab-vpnab. For VPN B the configuration installs the static route directly into the routing tables defined in the routing table group vpnb-vpnab.

Configure BGP between the PE and CE Routers

In this configuration example, the vpnab-site1 BGP group for VPN A installs the routes learned from the BGP session into the routing tables defined in the vpnab-vpnab routing table group. For VPN AB, the vpnab-site1 group installs the routes learned from the BGP session into the routing tables defined in the vpnab-vpna_and_vpnab routing table group. For VPN B, the vpnb-site1 group installs the routes learned from the BGP session into the routing tables defined in the vpnb-vpnab routing table group. Note that interface routes are not needed for this configuration.

The VRF import and export policies are similar to those defined in “Configure Static Routes between the PE and CE Routers” on page 185, except the export protocol is BGP instead of a static route. On all vrf-export policies, you use the from protocol bgp statement.

On Router PE1, configure BGP between the PE and CE routers as follows:

```
[edit]
routing-instances {
  VPN-A {
    instance-type vrf;
    interface fe-1/0/0.0;
    route-distinguisher 10.255.14.175:3;
    vrf-import vpn-a-import;
    vrf-export vpn-a-export;
    protocols {
      bgp {
        group vpn-a-site1 {
          family inet {
            unicast {
              rib-group vpn-a-vpnab;
            }
          }
        }
        peer-as 1;
        neighbor 192.168.197.141;
      }
    }
  }
  VPN-AB {
    instance-type vrf;
    interface fe-1/1/0.0;
    route-distinguisher 10.255.14.175:9;
    vrf-import vpnab-import;
    vrf-export vpnab-export;
    protocols {
      bgp {
        group vpnab-site1 {
          family inet {
            unicast {
              rib-group vpnab-vpn-a_and_vpn-b;
            }
          }
        }
        peer-as 9;
        neighbor 192.168.197.178;
      }
    }
  }
}
```

```

VPN-B {
  instance-type vrf;
  interface fe-1/0/2.0;
  route-distinguisher 10.255.14.175:10;
  vrf-import vpnb-import;
  vrf-export vpnb-export;
  protocols {
    bgp {
      group vpnb-site1 {
        family inet {
          unicast {
            rib-group vpnb-vpnab;
          }
        }
      }
      neighbor 192.168.197.242 {
        peer-as 10;
      }
    }
  }
}

```

Configure OSPF between the PE and CE Routers

In this configuration example, routes learned from the OSPF session for VPN A are installed into the routing tables defined in the `vpna-vpnab` routing table group. For VPN AB, routes learned from the OSPF session are installed into the routing tables defined in the `vpnab-vpna_and_vpnab` routing table group. For VPN B, routes learned from the OSPF session are installed into the routing tables defined in the `vpnb-vpnab` routing table group.

The VRF import and export policies are similar to those defined in “Configure Static Routes between the PE and CE Routers” on page 185 and “Configure BGP between the PE and CE Routers” on page 190, except the export protocol is OSPF instead of BGP or a static route. Therefore, on all `vrf-export` policies, you use the `from protocol ospf` statement instead of the `from protocol <static | bgp>` statement.

On Router PE1, configure OSPF between the PE and CE routers as follows:

```

[edit]
routing-instances {
  VPN-A {
    instance-type vrf;
    interface fe-1/0/0.0;
    route-distinguisher 10.255.14.175:3;
    vrf-import vpna-import;
    vrf-export vpna-export;
    protocols {
      ospf {
        rib-group vpna-vpnab;
        export vpna-import;
        area 0.0.0.0 {
          interface fe-1/0/0.0;
        }
      }
    }
  }
}

```

```

VPN-AB {
  instance-type vrf;
  interface fe-1/1/0.0;
  route-distinguisher 10.255.14.175:9;
  vrf-import vpnab-import;
  vrf-export vpnab-export;
  protocols {
    ospf {
      rib-group vpnab-vpna_and_vpnab;
      export vpnab-import;
      area 0.0.0.0 {
        interface fe-1/1/0.0;
      }
    }
  }
}
VPN-B {
  instance-type vrf;
  interface fe-1/0/2.0;
  route-distinguisher 10.255.14.175:10;
  vrf-import vpnb-import;
  vrf-export vpnb-export;
  protocols {
    ospf {
      rib-group vpnb-vpnab;
      export vpnb-import;
      area 0.0.0.0 {
        interface fe-1/0/2.0;
      }
    }
  }
}
}

```

Configure Static, BGP, and OSPF Routes between the PE and CE Routers

This section shows how to configure the routes between the PE and CE routers using a combination of static routes, BGP, and OSPF, as follows:

The connection between Router PE1 and Router CE1 uses static routing.

The connection between Router PE1 and Router CE2 uses BGP.

The connection between Router PE1 and Router CE3 uses OSPF.

Here, the configuration for VPN AB also includes a static route to CE1.

On Router PE1, configure a combination of static routing, BGP, and OSPF between the PE and CE routers:

```

[edit]
routing-instances {
  VPN-A {
    instance-type vrf;
    interface fe-1/0/0.0;
    route-distinguisher 10.255.14.175:3;
    vrf-import vpna-import;
    vrf-export vpna-export;
  }
}

```

```

routing-options {
  static {
    rib-group vpnab-vpnab;
    route 10.255.14.155/32 next-hop 192.168.197.141;
  }
}
VPN-AB {
  instance-type vrf;
  interface fe-1/1/0.0;
  route-distinguisher 10.255.14.175:9;
  vrf-import vpnab-import;
  vrf-export vpnab-export;
  protocols {
    bgp {
      group vpnab-site1 {
        family inet {
          unicast {
            rib-group vpnab-vpna_and_vpnab;
          }
        }
        export to-vpnab-site1;
        peer-as 9;
        neighbor 192.168.197.178;
      }
    }
  }
}
VPN-B {
  instance-type vrf;
  interface fe-1/0/2.0;
  route-distinguisher 10.255.14.175:10;
  vrf-import vpnb-import;
  vrf-export vpnb-export;
  protocols {
    ospf {
      rib-group vpnb-vpnab;
      export vpnb-import;
      area 0.0.0.1 {
        interface t3-0/3/3.0;
      }
    }
  }
}
policy-options {
  policy-statement to-vpnab-site1 {
    term a {
      from protocol static;
      then accept;
    }
    term b {
      from protocol bgp;
      then accept;
    }
    term c {
      then reject;
    }
  }
}

```

Configuring Overlapping VPNs Using Automatic Route Export

A problem with multiple routing instances is how to export routes between routing instances. This can be accomplished in JUNOS software by configuring routing table groups for each routing instance that needs to export routes to other routing tables. For information on how to configure overlapping VPNs using routing table groups, see “Configure Overlapping VPNs Using Routing Table Groups” on page 183.

However, using routing table groups has limitations:

- Routing table group configuration is complex. A unique routing table group must be defined for each routing instance that will export routes.

- You must also configure a unique routing table group for each protocol that will export routes.

To limit and sometimes eliminate the need to configure routing table groups in multiple routing instance topologies, you can use the functionality provided by the auto-export statement.

The auto-export statement is particularly useful for configuring overlapping VPNs—VPN configurations where more than one VRF lists the same community route target in its vrf-import policy. The auto-export statement finds out which routing tables to export routes from and import routes to by examining the existing policy configuration.

The auto-export statement automatically exports routes between the routing instances referencing a given route target community. When the auto-export statement is configured, a VRF target tree is constructed based on the vrf-import and vrf-export policies configured on the system. If a routing instance references a target in its vrf-import policy, it is added to the import list for the target. If it references a specific route target in its vrf-export policy, it is added to the export list for that target. Route targets where there is a single importer that matches a single exporter or with no importers or exporters are ignored.

Changes to routing tables that export route targets are tracked. When a route change occurs, the routing instance's vpn-export policy is applied to the route. If it is allowed, the route is imported to all the import tables (subject to the vrf-import policy) of the route targets set by the export policy.

The sections that follow describe how to configure overlapping VPNs using the auto-export statement for interinstance export in addition to routing table groups:

- Configuring Overlapping VPNs with BGP and auto-export on page 196

- Configuring Overlapping VPNs and Additional Tables on page 197

- Configuring auto-export for all VRF Instances on page 198

Configuring Overlapping VPNs with BGP and auto-export

The following example provides the configuration for an overlapping VPN where BGP is used between the PE and CE routers.

Configure routing instance VPN-A as follows:

```
[edit]
routing-instances {
  VPN-A {
    instance-type vrf;
    interface fe-1/0/0.0;
    route-distinguisher 10.255.14.175:3;
    vrf-import vpn-a-import;
    vrf-export vpn-a-export;
    routing-options {
      auto-export;
    }
    protocols {
      bgp {
        group vpn-a-site1 {
          peer-as 1;
          neighbor 192.168.197.141;
        }
      }
    }
  }
}
```

Configure routing instance VPN-AB as follows:

```
[edit]
routing-instances {
  VPN-AB {
    instance-type vrf;
    interface fe-1/1/0.0;
    route-distinguisher 10.255.14.175:9;
    vrf-import vpnab-import;
    vrf-export vpnab-export;
    routing-options {
      auto-export;
    }
    protocols {
      bgp {
        group vpnab-site1 {
          peer-as 9;
          neighbor 192.168.197.178;
        }
      }
    }
  }
}
```

For this configuration, the auto-export statement replaces the functionality that was provided by a routing table group configuration. However, sometimes additional configuration is required.

Since the vrf-import policy and the vrf-export policy from which the auto-export statement deduces the import and export matrix are configured on a per-instance basis, it is necessary to be able to enable or disable them for unicast and multicast, in case multicast network layer reachability information (NLRI) is configured.

Configuring Overlapping VPNs and Additional Tables

It might be necessary to use the auto-export statement between overlapping VPNs, but require that a subset of the routes learned from a VRF table be installed into the inet.0 table or in routing-instance.inet.2.

To support this type of scenario, where not all of the information needed is present in the vrf-import and vrf-export policies, you configure an additional list of routing tables using an additional routing table group.

To add routes from VPN-A and VPN-AB to inet.0 in the example described in “Configuring Overlapping VPNs with BGP and auto-export” on page 196, you need to include the following additional configuration statements:

Configure the routing options as follows:

```
[edit]
routing-options {
  rib-groups {
    inet-access {
      import-rib inet.0;
    }
  }
}
```

Configure routing instance VPN-A as follows:

```
[edit]
routing-instances {
  VPN-A {
    routing-options {
      auto-export {
        family inet {
          unicast {
            rib-group inet-access;
          }
        }
      }
    }
  }
}
```

Configure routing instance VPN-AB as follows:

```
[edit]
routing-instances {
  VPN-AB {
    routing-options {
      auto-export {
        family inet {
          unicast {
            rib-group inet-access;
          }
        }
      }
    }
  }
}
```

Routing table groups are used in this configuration differently from how they are generally used in JUNOS software. Routing table groups normally require that the exporting routing table be referenced as the primary import routing table in the routing table group. For this configuration, the restriction does not apply. The routing table group functions as an additional list of tables to which to export routes.

Configuring auto-export for all VRF Instances

The following configuration allows you to configure the auto-export statement for all of the routing instances in a configuration group:

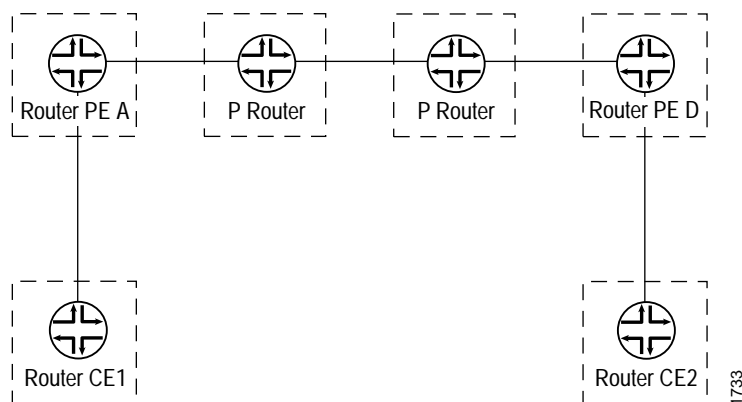
```
[edit]
groups {
  vrf-export-on {
    routing-instances {
      <*> {
        routing-options {
          auto-export;
        }
      }
    }
  }
}

apply-groups vrf-export-on;
```

Configure a GRE Tunnel Interface between PE Routers

This example shows how to configure a generic routing encapsulation (GRE) tunnel interface between PE routers to provide VPN connectivity. You can use this configuration to tunnel VPN traffic across a non-MPLS core network. The network topology used in this example is shown in Figure 24. Note that the provider (P) routers shown in this illustration do not run MPLS.

Figure 24: PE Router A and PE Router D Connected by a GRE Tunnel Interface



Configure the Routing Instance on Router A

Configure a routing instance on Router A as follows:

```
[edit routing-instances]
gre-config {
  instance-type vrf;
  interface fe-1/0/0.0;
  route-distinguisher 10.255.14.176:69;
  vrf-import import-config;
  vrf-export export-config;
  protocols {
    ospf {
      export import-config;
      area 0.0.0.0 {
        interface all;
      }
    }
  }
}
```

Configure the Routing Instance on Router D

Configure a routing instance on Router D as follows:

```
[edit routing-instances]
gre-config {
  instance-type vrf;
  interface fe-1/0/1.0;
  route-distinguisher 10.255.14.178:69;
  vrf-import import-config;
  vrf-export export-config;
  protocols {
    ospf {
      export import-config;
      area 0.0.0.0 {
        interface all;
      }
    }
  }
}
```

Configure MPLS, BGP, and OSPF on Router A

Although MPLS does not need to be configured on the P routers in this example, it is needed on the PE routers for the interface between the PE and CE routers and on the GRE interface (gr-1/1/0.0) linking the PE routers (Router A and Router D). Configure MPLS, BGP, and OSPF on Router A as follows:

```
[edit protocols]
mpls {
  interface all;
}
bgp {
  group pe-to-pe {
    type internal;
    neighbor 10.255.14.178 {
      family inet-vpn {
        unicast;
      }
    }
  }
}
ospf {
  area 0.0.0.0 {
    interface all;
    interface gr-1/1/0.0 {
      disable;
    }
  }
}
```

Configure MPLS, BGP, and OSPF on Router D

Although MPLS does not need to be configured on the P routers in this example, it is needed on the PE routers for the interface between the PE and CE routers and on the GRE interface (gr-1/1/0.0) linking the PE routers (Router D and Router A). Configure MPLS, BGP, and OSPF on Router D as follows:

```
[edit protocols]
mpls {
  interface all;
}
bgp {
  group pe-to-pe {
    type internal;
    neighbor 10.255.14.176 {
      family inet-vpn {
        unicast;
      }
    }
  }
}
```

```
ospf {
  traffic-engineering;
  area 0.0.0.0 {
    interface all;
    interface fxp0.0 {
      disable;
    }
    interface gr-1/1/0.0 {
      disable;
    }
  }
}
```

Configure the Tunnel Interface on Router A

Configure the tunnel interface on Router A as follows (note that the tunnel is unnumbered):

```
[edit interfaces interface-name]
unit 0 {
  tunnel {
    source 10.255.14.176;
    destination 10.255.14.178;
  }
  family inet;
  family mpls;
}
```

Configure the Tunnel Interface on Router D

Configure the tunnel interface on Router D as follows (note that the tunnel is unnumbered):

```
[edit interfaces interface-name]
unit 0 {
  tunnel {
    source 10.255.14.178;
    destination 10.255.14.176;
  }
  family inet;
  family mpls;
}
```

Configure the Routing Options on Router A

As part of the routing options configuration for Router A, you need to configure routing table groups to enable VPN route resolution in the inet.3 routing table.

Configure the routing options on Router A as follows:

```
[edit routing-options]
interface-routes {
  rib-group inet if-rib;
}
rib inet.3 {
  static {
    route 10.255.14.178/32 next-hop gr-1/1/0.0;
  }
}
```

```

rib-groups {
  if-rib {
    import-rib [ inet.0 inet.3 ];
  }
}

```

Configure the Routing Options on Router D

As part of the routing options configuration for Router D, you need to configure routing table groups to enable VPN route resolution in the inet.3 routing table.

Configure the routing options on Router D as follows:

```

[edit routing-options]
interface-routes {
  rib-group inet if-rib;
}
rib inet.3 {
  static {
    route 10.255.14.176/32 next-hop gr-1/1/0.0;
  }
}
rib-groups {
  if-rib {
    import-rib [ inet.0 inet.3 ];
  }
}

```

Configuration Summary for Router A

Configure the Routing Instance	<pre> gre-config { instance-type vrf; interface fe-1/0/0.0; route-distinguisher 10.255.14.176:69; vrf-import import-config; vrf-export export-config; protocols { ospf { export import-config; area 0.0.0.0 { interface all; } } } } </pre>
---------------------------------------	---

Configure MPLS	<pre> mpls { interface all; } </pre>
-----------------------	--

Configure BGP	<pre> bgp { traceoptions { file bgp.trace world-readable; flag update detail; } group pe-to-pe { type internal; neighbor 10.255.14.178 { family inet-vpn { unicast; } } } } </pre>
Configure OSPF	<pre> ospf { area 0.0.0.0 { interface all; interface gr-1/1/0.0 { disable; } } } </pre>
Configure the Tunnel Interface	<pre> interface-name { unit 0 { tunnel { source 10.255.14.176; destination 10.255.14.178; } family inet; } } </pre>
Configure Routing Options	<pre> interface-routes { rib-group inet if-rib; } rib inet.3 { static { route 10.255.14.178/32 next-hop gr-1/1/0.0; } } rib-groups { if-rib { import-rib [inet.0 inet.3]; } } </pre>

Configuration Summary for Router D

Configure the Routing Instance

```
gre-config {
  instance-type vrf;
  interface fe-1/0/1.0;
  route-distinguisher 10.255.14.178:69;
  vrf-import import-config;
  vrf-export export-config;
  protocols {
    ospf {
      export import-config;
      area 0.0.0.0 {
        interface all;
      }
    }
  }
}
```

Configure MPLS

```
mpls {
  interface all;
}
```

Configure BGP

```
bgp {
  group pe-to-pe {
    type internal;
    neighbor 10.255.14.176 {
      family inet-vpn {
        unicast;
      }
    }
  }
}
```

Configure OSPF

```
ospf {
  traffic-engineering;
  area 0.0.0.0 {
    interface all;
    interface fxp0.0 {
      disable;
    }
    interface gr-1/1/0.0 {
      disable;
    }
  }
}
```

Configure the Tunnel Interface

```
interface-name {
  unit 0 {
    tunnel {
      source 10.255.14.178;
      destination 10.255.14.176;
    }
    family inet;
  }
}
```

Configure the Routing Options

```

interface-routes {
  rib-group inet if-rib;
}
rib inet.3 {
  static {
    route 10.255.14.176/32 next-hop gr-1/1/0.0;
  }
}
rib-groups {
  if-rib {
    import-rib [ inet.0 inet.3 ];
  }
}

```

Configure a GRE Tunnel Interface between a PE and CE Router

This example shows how to configure a GRE tunnel interface between a PE router and a CE router. You can use this configuration to tunnel VPN traffic across a non-MPLS core network. The network topology used in this example is shown in Figure 25.

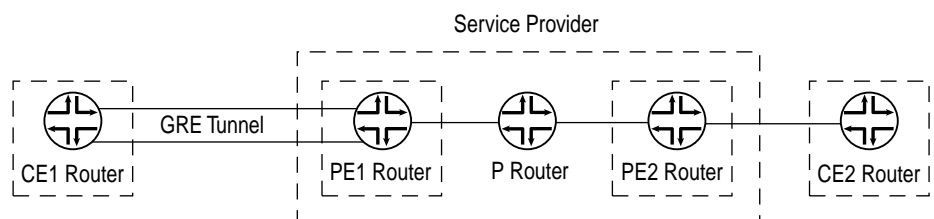
For this example, complete the procedures described in the following sections:

Configure the Routing Instance without the Encapsulating Interface on page 205

Configure the Routing Instance with the Encapsulating Interface on page 207

Configure the GRE Tunnel Interface on Router CE1 on page 208

Figure 25: GRE Tunnel between the CE Router and the PE Router



1744

Configure the Routing Instance without the Encapsulating Interface

You can configure the routing instance either with or without the encapsulating interface. The sections that follow describe how to configure the routing instance without it:

Configure the Routing Instance on Router PE1 on page 206

Configure the GRE Tunnel Interface on Router PE1 on page 206

Configure the Encapsulation Interface on Router PE1 on page 206

Configure the Routing Instance on Router PE1

Configure the routing instance on Router PE1 as follows:

```
[edit routing-instances]
vpna {
  instance-type vrf;
  interface gr-1/2/0.0;
  route-distinguisher 10.255.14.174:1;
  vrf-import vpna-import;
  vrf-export vpna-export;
  protocols {
    bgp {
      group vpna {
        type external;
        peer-as 100;
        as-override;
        neighbor 10.49.2.1;
      }
    }
  }
}
```

Configure the GRE Tunnel Interface on Router PE1

Configure the GRE tunnel interface on Router PE1 as follows:

```
[edit interfaces gr-1/2/0]
unit 0 {
  tunnel {
    source 192.168.197.249;
    destination 192.168.197.250;
  }
  family inet {
    address 10.49.2.2/30;
  }
}
```

In this example, interface t3-0/1/3 acts as the encapsulating interface for the GRE tunnel.

Configure the Encapsulation Interface on Router PE1

Configure the encapsulation interface on Router PE1 as follows:

```
[edit interfaces t3-0/1/3]
unit 0 {
  family inet {
    address 192.168.197.249/30;
  }
}
```


Configure the Routing Instance with the Encapsulating Interface

If the tunnel encapsulating interface, t3-0/1/3, is also configured under the routing instance, then you need to specify the name of that routing instance under the interface definition. The system uses this routing instance to search for the tunnel destination address.

To configure the routing instance with the encapsulating interface, complete the procedures described in the following sections:

Configure the Routing Instance on Router PE1 on page 207

Configure the GRE Tunnel Interface on Router PE1 on page 207

Configure the Encapsulation Interface on Router PE1 on page 208

Configure the Routing Instance on Router PE1

If you configure the tunnel encapsulating interface under the routing instance, then configure the routing instance on Router PE1 as follows:

```
[edit routing-instances]
vpna {
  instance-type vrf;
  interface gr-1/2/0.0;
  interface t3-0/1/3.0;
  route-distinguisher 10.255.14.174:1;
  vrf-import vpna-import;
  vrf-export vpna-export;
  protocols {
    bgp {
      group vpna {
        type external;
        peer-as 100;
        as-override;
        neighbor 10.49.2.1;
      }
    }
  }
}
```

Configure the GRE Tunnel Interface on Router PE1

Configure the GRE tunnel interface on Router PE1 as follows:

```
[edit interfaces gr-1/2/0]
unit 0 {
  tunnel {
    source 192.168.197.249;
    destination 192.168.197.250;
    routing-instance {
      destination vpna;
    }
  }
  family inet {
    address 10.49.2.2/30;
  }
}
```

Configure the Encapsulation Interface on Router PE1

Configure the encapsulation interface on Router PE1 as follows:

```
[edit interfaces t3-0/1/3]
unit 0 {
  family inet {
    address 192.168.197.249/30;
  }
}
```

Configure the GRE Tunnel Interface on Router CE1

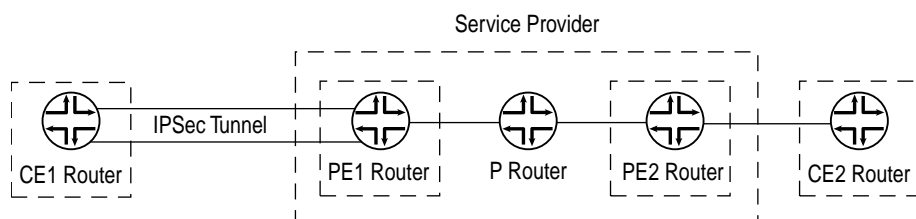
Configure the GRE tunnel interface on Router CE1 as follows:

```
[edit interfaces gr-1/2/0]
unit 0 {
  tunnel {
    source 192.168.197.250;
    destination 192.168.197.249;
  }
  family inet {
    address 10.49.2.1/30;
  }
}
```

Configure an ES Tunnel Interface between a PE and CE Router

This example shows how to configure an ES tunnel interface between a PE router and a CE router in a Layer 3 VPN. The network topology used in this example is shown in Figure 26.

Figure 26: ES Tunnel Interface (IPSec Tunnel) between the CE Router and the PE Router



1745

To configure this example, complete the steps in the following sections:

Configure IPSec on Router PE1 on page 209

Configure the Routing Instance without the Encapsulating Interface on page 209

Configure the Routing Instance with the Encapsulating Interface on page 210

Configure the ES Tunnel Interface on Router CE1 on page 212

Configure IPSec on Router CE1 on page 212

Configure IPsec on Router PE1

Configure IP Security (IPsec) on Router PE1 as follows:

```
[edit security]
ipsec {
  security-association sa-esp-manual {
    mode tunnel;
    manual {
      direction bidirectional {
        protocol esp;
        spi 16000;
        authentication {
          algorithm hmac-md5-96;
          key ascii-text "$9$ABULt1heK87dsWLDk.P3nrevM7V24ZHkPaZ/tpOcSvWLNwgZUH";
        }
        encryption {
          algorithm des-cbc;
          key ascii-text "$9$/H8Q90IyrvL7VKMZjHqQzcyleLN";
        }
      }
    }
  }
}
```

Configure the Routing Instance without the Encapsulating Interface

You can configure the routing instance on Router PE1 with or without the encapsulating interface (t3-0/1/3 in this example). The following sections describe how to configure the routing instance without it:

Configure the Routing Instance on Router PE1 on page 210

Configure the ES Tunnel Interface on Router PE1 on page 210

Configure the Encapsulating Interface for the ES Tunnel on Router PE1 on page 210

Configure the Routing Instance on Router PE1

Configure the routing instance on Router PE1 as follows:

```
[edit routing-instances]
vpna {
  instance-type vrf;
  interface es-1/2/0.0;
  route-distinguisher 10.255.14.174:1;
  vrf-import vpna-import;
  vrf-export vpna-export;
  protocols {
    bgp {
      group vpna {
        type external;
        peer-as 100;
        as-override;
        neighbor 10.49.2.1;
      }
    }
  }
}
```

Configure the ES Tunnel Interface on Router PE1

Configure the ES tunnel interface on Router PE1 as follows:

```
[edit interfaces es-1/2/0]
unit 0 {
  tunnel {
    source 192.168.197.249;
    destination 192.168.197.250;
  }
  family inet {
    address 10.49.2.2/30;
    ipsec-sa sa-esp-manual;
  }
}
```

Configure the Encapsulating Interface for the ES Tunnel on Router PE1

For this example, interface t3-0/1/3 is the encapsulating interface for the ES tunnel. Configure interface t3-0/1/3 as follows:

```
[edit interfaces t3-0/1/3]
unit 0 {
  family inet {
    address 192.168.197.249/30;
  }
}
```

Configure the Routing Instance with the Encapsulating Interface

If the tunnel encapsulating interface, t3-0/1/3, is also configured under the routing instance, you need to specify the routing instance name under the interface definition. The system uses this routing instance to search for the tunnel destination address for the IPSec tunnel using manual security association.

The following sections describe how to configure the routing instance with the encapsulating interface:

Configure the Routing Instance on Router PE1 on page 211

Configure the ES Tunnel Interface on Router PE1 on page 211

Configure the Encapsulating Interface on Router PE1 on page 212

Configure the Routing Instance on Router PE1

Configure the routing instance on Router PE1 (including the tunnel encapsulating interface) as follows:

```
[edit routing-instances]
vpna {
  instance-type vrf;
  interface es-1/2/0.0;
  interface t3-0/1/3.0;
  route-distinguisher 10.255.14.174:1;
  vrf-import vpna-import;
  vrf-export vpna-export;
  protocols {
    bgp {
      group vpna {
        type external;
        peer-as 100;
        as-override;
        neighbor 10.49.2.1;
      }
    }
  }
}
```

Configure the ES Tunnel Interface on Router PE1

Configure the ES tunnel interface on Router PE1 as follows:

```
[edit interfaces es-1/2/0]
unit 0 {
  tunnel {
    source 192.168.197.249;
    destination 192.168.197.250;
    routing-instance {
      destination vpna;
    }
  }
  family inet {
    address 10.49.2.2/30;
    ipsec-sa sa-esp-manual;
  }
}
```

Configure the Encapsulating Interface on Router PE1

Configure the encapsulating interface on Router PE1 as follows:

```
[edit interfaces t3-0/1/3]
unit 0 {
  family inet {
    address 192.168.197.249/30;
  }
}
```

Configure the ES Tunnel Interface on Router CE1

Configure the ES tunnel interface on Router CE1 as follows:

```
[edit interfaces es-1/2/0]
unit 0 {
  tunnel {
    source 192.168.197.250;
    destination 192.168.197.249;
  }
  family inet {
    address 10.49.2.1/30;
    ipsec-sa sa-esp-manual;
  }
}
```

Configure IPsec on Router CE1

Configure IPsec on Router CE1 as follows:

```
[edit security]
ipsec {
  security-association sa-esp-manual {
    mode tunnel;
    manual {
      direction bidirectional {
        protocol esp;
        spi 16000;
        authentication {
          algorithm hmac-md5-96;
          key ascii-text "$9$ABULt1heK87dsWLDk.P3nrevM7V24ZHkPaZ/tp0cSvWLNwgZUH";
        }
        encryption {
          algorithm des-cbc;
          key ascii-text "$9$/H8Q90IyrvL7VKMZjHqQzcyleLN";
        }
      }
    }
  }
}
```

Configure SCU and DCU for Layer 3 VPNs

For information on how to configure source class usage (SCU) for a Layer 3 VPN loopback interface, see the *JUNOS Internet Software Configuration Guide: Network Management*.

For information on how to configure SCU and destination class usage (DCU) to count packets on Layer 3 VPNs, see the *JUNOS Internet Software Configuration Guide: Interfaces and Class of Service*.

.....